Connection (Connection)

Properties

ConnectionString – Connection string

`BSTR ConnectionString`

C++ Type	VB Type	R/W	Value range	Description
BSTR	String	Read-Only	valid strings	Defines connection device, protocol etc.

Returns the ConnectionString for the current connection. All characters of this string are always upper case. If the string is not empty (length ≠ 0) the connection does exist. On disconnection the string is reset to 0 length.

LibraryVersion – Version number

`BSTR LibraryVersion`

C++ Type	VB Type	R/W	Value range	Description
BSTR	String	Read-Only	#.#.#.# where # is a number 0...9999	Version of this program library

Returns the version of the program library that is used.

Methods

Connect – Establish connection to FM

`Connect(ConnectionString)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
ConnectionString	BSTR	String	In	valid strings	Defines connection device, protocol etc.

Establishes connection to the FM and loads all properties.

ConnectionString contains the necessary information to establish the connection. It consists of the connection device definition and the protocol name. This version of the library supports serial communication via the COM-Ports using the MLPV6 protocol only. So the only valid syntax looks as follows: `COMPORT=#; PROTOCOL=MLPV6` while # represents the number of the COM-Port. The fields have to be separated by a semicolon (;). The MLPV6 protocol is set as default so that the `PROTOCOL` field can be omitted. Examples for valid ConnectionStrings (ConnectionString is NOT case sensitive):

```
COMPORT=1; PROTOCOL=MLPV6
COMPORT=4
COMPORT=12; PORTOCOL=MLPV6
```

Disconnect – Disconnect from FM

`Disconnect()`

After disconnection all properties are invalid.

Events

OnDieCoverClose – Die cover closed

OnDieCoverClosed

Triggered when the FM's Die cover is closed. The FM always goes to QUIET state (not ready for Franking) when the Die cover is opened and can not be set ready for Franking until the Die cover is closed.

OnDieCoverOpen – Die cover opened

OnDieCoverOpen

Triggered when the FM's Die cover is opened. The FM always goes to QUIET state (not ready for Franking) when the Die cover is opened and can not be set ready for Franking until the Die cover is closed.

OnDisconnect – Undexpected disconnection

OnDisconnect

Triggered when the connection between PC and FM is unexpectedly broken.

OnFranking – A Franking is released

OnFranking(Postage, JobRest, StatusCode)

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Postage	CURRENCY	Currency	In	any	Postage value Franked
JobRest	short	Integer	In	any	Remaining number of items in this Preselection session. -1 if no Preselection was chosen on SetDecades(Tax).
StatusCode	RETVALS	RETVALS	In	R_OK, E_NOK, R_QUIET, all W_..., all L_...	Returnes information about the state of the machine.

This event is triggered every time a Franking is released on the FM. This may also happen without the previous call of SetDecades, as the user may perform this action via the FM keyboard. For Autotax applications the postage represents the sum of Autotax and Acquisitiontax. See also *SetDecades – Set FM ready for Franking* on page 16, *SetDecadesTax – Set Autotax FM ready for Franking (for Autotax applications only)* on page 17 and *RETVALS* on page 26.

OnNoMoreTapes – No more Tapes

OnNoMoreTapes

Triggered when in Franking Mode FRK_TAPE and there are no more Tapes in the FM. Franking can be continued by calling TapesPresent. See also *TapesPresent – Tapes present again* on page 17.

OnQuiet – FM goes to QUIET state (not ready for Franking)

`OnQuiet(StatusCode)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
StatusCode	RETVALS	RETVALS	In	R_OK, E_HSB_KEY, E_HSB_HOT, E_HSB_STANDBY, E_HSB_INKCOVER	Returnes information about the state of the machine.

Triggered when the FM goes to QUIET state without a corresponding command (GoQuiet) from the PC.

OnRotorError – Rotor error

`OnRotorError(StatusCode)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
StatusCode	RETVALS	RETVALS	In	E_CALL_SERVICE, E_ROTOR, E_CROSS, E_BASE_SPEED, E_DECADE, E_PRINTER	Returnes information about the state of the machine.

Triggered when the FM has a problem that does not allow further Frankings. This may be a problem setting decade wheels or the rotor position etc. See also *RETVALS* on page 26.

OnTMSEnd – End of TMS procedure

`OnTMSEnd(Descending, Message, StatusCode)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Descending	CURRENCY	Currency	In	Any	New value of descending counter.
Message	BSTR	String	In	up to 32 chars	Message from TMS center. Empty string if no message was received.
StatusCode	RETVALS	RETVALS	In	R_OK, E_DESC_MAX, E_SUM_OFL, E_AMOUNT_MAX, E_AMOUNT_STEP, E_AMOUNT_MIN, E_TMS_ABORT, E_TMS_TRANSM, E_TMS_CONNECT, E_TMS_MODEM, E_TMS_LINE_BUSY, E_TMS_MODEM_INIT, E_TMS_FAILED, E_FMSTATE	Returnes information about the state of the machine.

Informs that the TMS procedure has finished either successful carried out or with an error. In the case of an error the message received from the TMS center may contain additional information about the cause of the error. If one calls the function to start the TMS process on an FM that does not support TMS, this event will be triggered with StatusCode = E_TMS_FAILED.

The TMS procedure can be started by calling the function TMSStart. See also *TMSStart – Start TMS reset* on page 18.

See also *RETVALS* on page 26.

Configuration (FMConfig)

Important: Before any of the properties or methods of FMConfig can be used the property ActiveConnection must be set.

Properties

The following applies to all properties except for ActiveConnection:
All properties are READ-ONLY. The properties are loaded from the FM when the connection is established. Non of the properties does ever change, so there is no need to reload them. On disconnection all properties become invalid.

ActiveConnection – Relation to Connection

`IConnection ActiveConnection`

C++ Type	VB Type	R/W	Value range
IConnection*	Connection	Read-Write	Any

Has to be set before the first use of any property or method of FMConfig. It defines the Connection that FMConfig uses to perform the functions in the FM.

AcquisitonTaxAvailable – Availability of Acquisitiontax

`VARIANT_BOOL AcquisitionTaxAvailable`

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

Determinates whether the FM features Acquisitiontax. Acquisitiontax is possible on Autotax FMs only. See also *SetDecadesTax – Set Autotax FM ready for Franking (for Autotax applications only)* on page 17.

AutoDate – Presence of Automatic Date

`VARIANT_BOOL AutoDate`

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

Determinates whether the FM has automatic date wheels.

AutoTaxFM – Autotax / Standard FM

`VARIANT_BOOL AutoDate`

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

TRUE if FM is a special Autotax FM. FALSE if FM is a standard (non Autotax) FM. This property allows deciding whether SetDecadesTax or SetDecades must be used to set the FM ready for Franking. The following table shows which function must be used for which FM application:

FM	Function to set ready for Franking
Autotax	SetDecadesTax
Standard	SetDecades

BaseModel – Base model

`BASES BaseModel`

C++ Type	VB Type	R/W	Value range
BASES	BASES	Read-Only	All of BASES

See also *BASES* on page 25.

BaseSWVer – Base software version

`BSTR BaseSWVer`

C++ Type	VB Type	R/W	Value range
BSTR	String	Read-Only	Any text

DecadeNumber – Number of decade wheels

`short DecadeNumber`

C++ Type	VB Type	R/W	Value range
short	Integer	Read-Only	4 or 5

See also *Examples for print image configuration* on page 9.

DecPointPosition – Position of decimal point

`short DecPointPosition`

C++ Type	VB Type	R/W	Value range
short	Integer	Read-Only	0..5

Position of decimal point in money values, right aligned. This property is necessary to format money values correctly to match FM's configuration. It may be used to generate an entry mask on the user interface. FixedZeros do also count as a position. See also *Examples for print image configuration* on page 9.

FixedZeros – Number of fixed zeros

`short FixedZeros`

C++ Type	VB Type	R/W	Value range
short	Integer	Read-Only	0..3

Number of fixed zeros after the decade wheels. This property is necessary to format money values correctly to match FM's configuration. It may be used to generate an entry mask on the user interface. See also *Examples for print image configuration* on page 9.

FMSWVer – FM software version

`BSTR FMSWVer`

C++ Type	VB Type	R/W	Value range
BSTR	String	Read-Only	Any

FMType – FM type

`FMTYPES FMType`

C++ Type	VB Type	R/W	Value range
FMTYPES	FMTYPES	Read-Only	All of FMTYPES

Determinates what type of FM is connected to the PC. See also *FMTYPES* on page 25.

FrankModeAvailItem – Availability of Franking Mode FRK_ITEMS

`VARIANT_BOOL FrankModeAvailItem`

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

Determinates whether the Item Franking Mode (FRK_ITEM) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see *SetDecades – Set FM ready for Franking* on page 16.

FrankModeAvailLetter – Availability of Franking Mode FRK_LETTER

```
VARIANT_BOOL FrankModeAvailLetter
```

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

Determinates whether the Letter Franking Mode (FRK_LETTER) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see *SetDecades – Set FM ready for Franking* on page 16.

FrankModeAvailNorm – Availability of Franking Mode FRK_NORM

```
VARIANT_BOOL FrankModeAvailNorm
```

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

Determinates whether the Normal Franking Mode (FRK_NORM) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see *SetDecades – Set FM ready for Franking* on page 16.

FrankModeAvailTape – Availability of Franking Mode FRK_TAPE

```
VARIANT_BOOL FrankModeAvailTape
```

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

Determinates whether the Tape Franking Mode (FRK_TAPE) is available. One of the available Franking Modes has to be given as parameter when calling the SetDecades function. The available Franking Modes depend on the Base model. For more details on Franking Modes see *SetDecades – Set FM ready for Franking* on page 16.

LastDecadeType – Type of the last decade wheel

```
LASTDECTYPES LastDecadeType
```

C++ Type	VB Type	R/W	Value range
LASTDECTYPES	LASTDECTYPES	Read-Only	LD09, LD05

Determinates whether the last decade wheel can have all values (0...9 = LD09) or only the values 0 and 5 (= LD05). This property is necessary to format money values correctly to match FM's configuration. It may be used to generate an entry mask on the user interface. See *Examples for print image configuration* on page 11 and LASTDECTYPES on page 25.

MailClassCylinderPositions – Number of positions on Mail Class cylinder

`SAFEARRAY(BSTR) MailClassTexts`

C++ Type	VB Type	R/W	Value range
short	Interger	Read-Only	0, 4, 5

Often not all positions on the Mail Class cylinder are in use. So there are less Mail Classes available than positions on the cylinder. This value is important for MailClassAdjust as it is possible that the printed Mail Class is one of the unused positions on the cylinder.
See also *SettableMailClasses – Number of available Mail Classes* on page 9 and *MailClassAdjust – Adjust Mail Class assignment* on page 15.

MailClassTexts – Texts assigned to Mail Classes

`SAFEARRAY(BSTR) MailClassTexts`

C++ Type	VB Type	R/W	Value range
SAFEARRAY(BSTR)	String()	Read-Only	One-dimensional array of up to 5 texts of max. 10 chars

Each settable Mail Class has a text assigned that is shown on the FM display when a Mail Class is chosen using the FMs keyboard. These texts usually correspond to the Mail Class Dies. Each text is max. 10 chars long. There can be up to 5 texts as there are up to 5 positions on Mail Class cylinder. Possibly not all cylinder positions are available for printing. So this array contains as many texts as there are settable Mail Classes. If there are zero settable Mail Classes the array is empty (no elements). To set the Mail Class for Franking, it is necessary to pass the index of the Mail Class. The index starts from 1 to the number of settable Mail Classes. See also *SettableMailClasses – Number of available Mail Classes* on page 9.

MarginAvailable – Availability of margin

`VARIANT_BOOL MarginAvailable`

C++ Type	VB Type	R/W	Value range
VARIANT_BOOL	Boolean	Read-Only	TRUE/FALSE

The margin indicates the distance between the edge of the letter and the beginning of the print. Margin can be set (is available) on FMs with optical release only.

SerialNr – Serial number of FM

`BSTR SerialNr`

C++ Type	VB Type	R/W	Value range
BSTR	String	Read-Only	up to 10 chars

Serial number that identifies the FM.

SettableMailClasses – Number of available Mail Classes

```
short SettableMailClasses
```

C++ Type	VB Type	R/W	Value range
short	Integer	Read-Only	0..5

Number of Mail Classes that are available. FMs that do not have Mail Classes return 0 (zero).

Examples for print image configuration

The labels of column 2 through 5 are the names of the properties, which specify the print image configuration.

Mask	DecadeNumber	DecPointPosition	FixedZeros	LastDecadeType
99.999	5	4	0	LD09
999.90	4	3	1	LD09
999.95	5	3	0	LD05
999.99	5	3	0	LD09
999.99(	5	4	1	LD09
9995	4	1	0	LD05
9999	4	0	0	LD09
9999.90(	5	4	2	LD09
99990	4	0	1	LD09
99999	5	0	0	LD09
999500	4	0	2	LD05
9999900	5	0	2	LD09

Meaning of column 'Mask':

9	Can be anything 0..9
5	Can be 0 or 5
0	Fixed Zero

Methods

DescRegHide – Hide (do not show) Descending Register

`DescRegHide()`

The Descending Register contains the amount of money that is currently stored inside the FM. This value can be shown on the FM display. This function hides the Descending Register on FM display.

DescRegShow – Show Descending Register

`DescRegShow()`

The Descending Register contains the amount of money that is currently stored inside the FM. This function shows the Descending Register on FM display.

FrankMenuLock – Lock the Franking Menu

`FrankMenuLock()`

Locks the Franking Menu so that Decade setting and Franking can not be done from FM keyboard anymore. All other functions stay available from FM keyboard.

FrankMenuUnlock – Unlock the Franking Menu

`FrankMenuUnlock()`

Unlocks the Franking Menu so that Decade setting and Franking are available directly from FM keyboard again.

HVLimitActivate – Activate High Value limit

`HVLimitActivate()`

Activates High Value limit so that Frankings with a higher value than specified can only be carried out after confirmation.

HVLimitDeactivate – Activate High Value limit

`HVLimitDeactivate()`

Deactivates High Value limit so that all Frankings can be carried out without confirmation.

KeyboardLock – Lock the FM keyboard

`KeybpardLock()`

Locks the FM keyboard so that it is accessible via the PC interface only.

KeyboardUnlock – Unlock the FM keyboard

```
KeybpardUnlock()
```

Unlocks the FM keyboard so that it is accessible via the PC interface and from the FM keyboard at the same time.

UserTimeoutActivate – Activate User Timeout

```
UserTimeoutActivate()
```

Activates the User Timeout. After a specified time of inactivity the FM goes automatically into sleep mode (turns off Base motor and display and goes into QUIET state). The duration of the User Timeout is defined in the FM software and can be changed by the service technician.

UserTimeoutDeactivate – Deactivate User Timeout

```
UserTimeoutDeactivate()
```

Deactivates the User Timeout. The FM never goes automatically into sleep mode (turns off Base motor and display and goes into QUIET state).

WarningLowCreditActivate – Activate Low Credit warning

```
UserTimeoutActivate()
```

Activates the Low Credit warning. If the Descending register drops below a specified value the user receives a warning. The warning level is defined in the FM software and can be changed by the service technician.

WarningLowCreditDeactivate – Deactivate Low Credit warning

```
UserTimeoutDeactivate()
```

Deactivates the Low Credit warning. No warning is issued concerning the Descending register.

Actions (FMActions)

Important: Before any of the methods of FMActions can be used the property ActiveConnection must be set.

Properties

ActiveConnection – Relation to Connection

`IConnection ActiveConnection`

C++ Type	VB Type	R/W	Value range
IConnection*	Connection	Read-Write	Any

Has to be set before the first use of any method of FMActions. It defines the Connection that FMActions uses to perform the functions in the FM.

Methods

BatchCounterActivate – Activates batch counter

`BatchCounterActivate()`

Activates the batch counter. The batch counter counts all non zero postage Frankings while it is active. It can only be stopped by clearing (see below).

BatchCounterClear – Read, clear and deactivate batch counter

`BatchCounterClear(On, Items, Value)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
On	VARIANT_BOOL	Boolean	Out	TRUE/FALSE	TRUE if batch counter is (was) active
Items	long	Long	Out	Any	Number of items franked since batch counter was activated
Value	CURRENCY	Currency	Out	Any	Total value franked since batch counter was activated

Reads, clears and deactivates batch counter. The items and the money value of the batch counter, are returned as well as the information whether the batch counter was active before this method was called. The batch counter is deactivated and its values are reset to zero respective FALSE.

BatchCounterRead – Read batch counter

`BatchCounterClear(On, Items, Value)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
On	VARIANT_BOOL	Boolean	Out	TRUE/FALSE	TRUE if batch counter is active
Items	long	Long	Out	Any	Number of items franked since batch counter was activated
Value	CURRENCY	Currency	Out	Any	Total value franked since batch counter was activated

Items and the money value of the batch counter, are returned as well as the information whether the batch counter is active.

GetCounterValues – Read postal counters

`GetCounterValues(Ascending, Descending, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Ascending	CURRENCY	Currency	Out	Any	Value of the Ascending counter
Descending	CURRENCY	Currency	Out	Any	Value of the Descending counter
Items	long	Long	Out	Any	Value of the Item counter

The current values of the postal counters are returned.

GetTimeDate – Current time and date of the FM

```
DATE GetTimeDate
```

C++ Type	VB Type	I/O/R	Value range	Description
DATE	Date	Ret	Any date	Current FM system time and date.

The current system time and date (local) of the FM.

GoQuiet – Set FM to QUIET state (not ready for Franking)

```
GoQuiet
```

Sets the FM to QUIET state. In QUIET state the machine is not ready for Franking and the decade wheels are not set.

HVLimitAbort – Not accept postage above High Value limit

```
RETVALS HVLimitAbort()
```

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
RetVal	RETVALS	RETVALS	Ret	R_OK, E_TIMEOUT	Returnes information about the state of the machine.

A SetDecades method can return with a RetVal = R_HVLIM saying that the given postage is higher than the High Value limit. This means that the decades are not set yet and the FM is not ready for Franking yet. To carry out the SetDecades command a confirmation is necessary. With HVLimitAbort the SetDecades command is aborted, meaning the high postage is not accepted.

One of the methods HVLimitAbort or HVLimitDeblock has to be called before the timeout (for most FMs around 8 sec.) has run down. The timeout starts to run when SetDecades returns R_HVLIM. If the HVLimitAbort is called too late, E_TIMEOUT is returned and the FM goes to QUIET state. See also *SetDecades – Set FM ready for Franking* on page 16 and *RETVALS* on page 25.

HVLimitDeblock – Accept postage above High Value limit

```
RETVALS HVLimitDeblock()
```

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
RetVal	RETVALS	RETVALS	Ret	R_OK, E_TIMEOUT	Returnes information about the state of the machine.

A SetDecades method can return with a RetVal = R_HVLIM saying that the given postage is higher than the High Value Limit. This means that the decades are not set yet and the FM is not ready for Franking yet. To carry out the SetDecades command a confirmation is necessary. With HVLimitDeblock the SetDecades command is carried out, meaning that the high postage is accepted.

One of the methods HVLimitAbort or HVLimitDeblock has to be called before the timeout (for most FMs around 8 sec.) has run down. The timeout starts to run when SetDecades returns R_HVLIM. If the HVLimitDeblock is called too late, E_TIMEOUT is returned and the FM goes to QUIET state. See also *SetDecades – Set FM ready for Franking* on page 16 and *RETVALS* on page 25.

MailClassAdjust – Adjust Mail Class assignment

```
MailClassAdjust(PrintedMailClass)
```

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
PrintedMailClass	short	Integer	In	1...MailClassCylinderPositions	Index of the printed Mail Class

Under very rare circumstances it can happen that the Mail Class assignment between the Mail Class texts and the positions of the Mail Class cylinder are shifted. Because of that, the wrong Mail Class is stamped. With this function it is possible to reassign the Mail Classes. The parameter PrintedMailClass has to be the index of the wrongly printed Mail Class. So the current position of the Mail Class cylinder will be reassigned with the Mail Class text of the given index. See also *MailClassCylinderPositions – Number of positions on Mail Class cylinder* on page 8 and *MailClassCylinderPositions – Number of positions on Mail Class cylinder* on page 8.

Example:

The following Mail Class texts are available:

Text	Index
Ground	1
Express	2
Airmail	3
(empty)	4

Franking with Mail Class:

Chosen:	Ground
Printed:	Airmail

Adjustment:

MailClassAdjust(3)

NewDateConfirm – Confirm new date request

```
NewDateConfirm
```

After the FM has requested a new print date (SetDecades(Tax) returned R_NEWDATE) this function needs to be called to confirm the request. On FMs with automatic date the date wheels are adjusted automatically as soon as this function is called. FMs without automatic date assume that the date wheels have been adjusted manually when NewDateConfirm is called.

SetDecades – Set FM ready for Franking

`RETVALS SetDecades(Postage, DPT, MailClass, FrankMode, Preselection, Margin)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Postage	CURRENCY	Currency	In	0…max. of print image configuration	Postage value to be Franked.
DPT	Long	Long	In	0 or any existing DPT number	Department to book Franking to (0 if no DPTs exist). If DPTs exist a valid number has to be given (not 0)!
MailClass	Short	Integer	In	1…(SettableMailClasses)	Index of MailClass to be printed (ignored if no MailClass is settable)
FrankMode	FRANKMODES	FRANKMODES	In	all available of FRANKMODES	Franking mode. Check properties for availability. If FRK_NORM is set, Preselection must be 0!
Preselection	Short	Integer	In	0…999	Max. number of items to be Franked with this settings (0 = no limit). If Preselection is 0, FrankMode must be FRK_NORM!
Margin	Short	Integer	In	1…255	Margin from the edge of the envelope to the beginning of the print. Available on FMs with optical release only!!! Value is ignored if Margin is not available. Check property for availability.
RetVal	RETVALS	RETVALS	Ret	R_OK, R_HVLIM, R_QUIET, R_NEWDATE, all W_..., all L_...	Returnes information about the state of the machine.

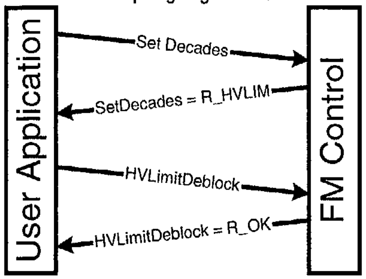
This is probably the most important function. It sets the FM ready for Franking by setting the postage value along with a range of other settings.

For each Franking that is released the OnFranking event is called. See *OnFranking – A Franking is released* on page 3.

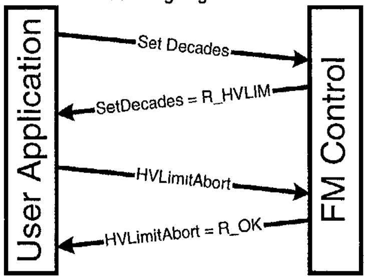
This function can NOT be used for Autotax FMs. See also *AutoTaxFM – Autotax / Standard FM* on page 5.

If the RetVal is R_HVLIM, the postage given is higher than the defined High Value Limit (and HV limit is active). After that, it is necessary to call either HVLimitAbort to abort the decade setting or HVLimitDeblock to accept the high postage. With HVLimitDeblock the decade wheels are set for one Franking only. After the Franking the FM goes back into QUIET state and the SetDecades function has to be repeated. See also *HVLimitAbort – Not accept postage above High Value limit* on page 14, *HVLimitDeblock – Accept postage above High Value limit* on page 14 and the following sequence diagrams.

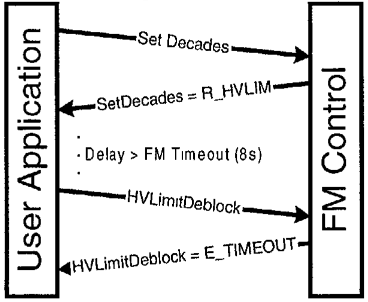
Accepting High Value



Declining High Value



Sending answer too late



For more information about allowed postage values see *Examples for print image configuration* on page 9.

For more information about Mail Classes *SettableMailClasses – Number of available Mail Classes* on page 9 and *MailClassTexts – Texts assigned to Mail Classes* on page 8.

For more information about Franking Modes see

FRANKMODES on page 25, *FrankModeAvailItem – Availability of Franking Mode FRK_ITEMS* on page 7 and the three following.
The relation between the value of margin and the real distance on the printed item depends on several other factors like Base speed etc.
More detailed information can be found in chapter 10 of the Service Manual for PLUS FMs. See also *MarginAvailable – Availability of margin* on page 8.
For all RetVals apart form R_OK and R_HVLIM the FM goes back into QUIET state (not ready for Franking). See also *RETVALS* on page 25.

SetDecadesTax – Set Autotax FM ready for Franking (for Autotax applications only)

`RETVALS SetDecadesTax(Autotax, Acquisitiontax, DPT, Margin)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Autotax	CURRENCY	Currency	In	0…max. of print image configuration	Autotax
Acquisitontax	CURRENCY	Currency	In	0…max. of print image configuration	Acquisitiontax
DPT	Long	Long	In	0 or any existing DPT number	Department to book Franking to (0 if no DPTs exist). If DPTs exist a valid number has to be given (not 0)!
Margin	Short	Integer	In	1…255	Margin from the edge of the envelope to the beginning of the print. Available on FMs with optical release only!!! Value is ignored if Margin is not available. Check property for availability.
RetVal	RETVALS	RETVALS	Ret	R_OK, R_HVLIM, R_QUIET, R_NEWDATE, all W_..., all L_...	Returnes information about the state of the machine.

SetDecadesTax can NOT be used for standard (non Autotax) FMs. See also *AutoTaxFM – Autotax / Standard FM* on page 5.
This function does the same as SetDecades but for Autotax applications only. It sets the FM ready for Franking by setting the Autotax and Acquisitiontax value along with other settings. The sum of Autotax and Acquisitiontax is printed. For Autotax applications that do not feature Acquisitiontax the Acquisitiontax value has to be set to zero. See also *AcquisitonTaxAvailable – Availability of Acquisitiontax* on page 5.
Mail Classes, Franking Modes and Preselection are not available for Autotax applications. Apart from that, the behavior of SetDecadesTax SetDecades is identical.
See also *SetDecades – Set FM ready for Franking* on page 16.
For all RetVals apart form R_OK and R_HVLIM the FM goes back into QUIET state (not ready for Franking). See also *RETVALS* on page 25.

TapesPresent – Tapes present again

`TapesPresent`

Can be sent to continue Franking Tapes after OnNoMoreTapes has been executed. See also *OnNoMoreTapes – No more Tapes* on page 3.

TextToDisplay – Write text to FM display

`TextToDisplay(Text, StartPos)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Text	BSTR	String	In	Any text with max. length = 32	Text to display on the FM
StartPos	Short	Integer	In	0…31	Position where the text starts

Displays any text on the FM display. If the text goes beyond the display the an exception with the ID INCORRDATA is raised. The text on the FMs display may be overwritten by "*" if User Timeout is active. The positions on the FM display are numbered as follows:

0															15
16															31

TMSStart – Start TMS reset

`TMSStart(Amount)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Amount	CURRENCY	Currency	In	any	Amount of money to load into the FM. The value range depends on the FM's configuration.

Starts the TMS process. The FM must be configured for TMS (center phone number, account number etc.) before calling this method. This function will return immediately so that the user application is not blocked while the TMS process is running. When the TMS process finishes the event OnTMSEnd is triggered. The whole TMS process can take up to about 150 sec. It is recommended that the user application starts a timer of about 3 minutes upon calling TMSStart. If the timer runs down without OnTMSEnd triggered it can be assumed that an error occurred (e.g. connection between PC and FM broken).

See also *OnTMSEnd – End of TMS procedure* on page 4.

Departments (FMDPT)

Important: Before any of the properties or methods of FMDPT can be used the property ActiveConnection must be set.

Properties

The following applies to all properties except for ActiveConnection:
All properties are READ-ONLY. The properties are loaded from the FM when the connection is established. Non of the properties does ever change, so there is no need to reload them. On disconnection all properties become invalid.

ActiveConnection – Relation to Connection

`IConnection ActiveConnection`

C++ Type	VB Type	R/W	Value range
IConnection*	Connection	Read-Write	Any

Has to be set before the first use of any property or method of FMDPT. It defines the Connection that FMDPT uses to perform the functions in the FM.

NrAccounts – Max. number of DPTs

`short ActiveConnection`

C++ Type	VB Type	R/W	Value range
short	Integer	Read-Only	0...max. Number of departments (currently = 800)

Maximal number of department accounts that can be opened in the FM.

NrDigits – Length of department numbers

`short ActiveConnection`

C++ Type	VB Type	R/W	Value range
short	Integer	Read-Only	Any

Max. number of digits a department number can have.

Methods

Clear – Read and clear one department

`Clear(DPT, Value, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
DPT	long	Long	In	Number of a existing department	Number of the department to be cleared
Value	CURRENCY	Currency	Out	any	Value Franked to this department
Items	long	Long	Out	any	Number of items Franked to this department

Reads the value and the number of items Franked to a department and resets them to zero.

ClearAll – Read total and clear all departments

`ClearAll(Value, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Value	CURRENCY	Currency	Out	any	Sum of all values Franked to departments
Items	long	Long	Out	any	Sum of items Franked to departments

Reads the total value and the number of items Franked to departments and resets all departments to zero.

Close – Close department

`RETVALS Close(DPT)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
DPT	long	Long	In	Number of a existing department	Number of the department to be closed
RetVal	RETVALS	RETVALS	Ret	R_OK, E_NOTCLEAR	Returnes information about the state of the machine.

Closes an existing department. The department must be clear to be closed. See also *RETVALS* on page 25.

Open – Open new department

`RETVALS Open(DPT)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
DPT	long	Long	In	Number of a not existing department	Number of the department to be opened
RetVal	RETVALS	RETVALS	Ret	R_OK, E_EXIST, E_FULL	Returnes information about the state of the machine.

Opens a new department with the given number. The department must not exist yet. DPT must not be longer than specified by the NrDigits property. See also *NrDigits – Length of department numbers* on page 19 and *RETVALS* on page 25.

Read – Read one department

`Read(DPT, Value, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
DPT	long	Long	In	Number of a existing department	Number of the department to be read
Value	CURRENCY	Currency	Out	Any	Value Franked to this department
Items	long	Long	Out	Any	Number of items Franked to this department

Reads the value and the number of items Franked to a department.

ReadAll – Read all departments

`ReadAll(DPT, Value, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
DPT	SAFEARRAY(long)	Long()	Out	Number of an existing department	One-dimensional array of department numbers
Value	SAFEARRAY(CURRENCY)	Currency()	Out	Any	One-dimensional array of values Franked to departments
Items	SAFEARRAY(long)	Long()	Out	Any	One-dimensional array of number of items Franked departments

Provides a table containing all open departments. For each department the department number, the value Franked to the department and the items Franked to the department are supplied.

All arrays contain the same number of elements. The elements with the same index belong together and represent the data of one department.

ReadTotal – Read total of all departments

`ReadNext(Value, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Value	CURRENCY	Currency	Out	Any	Sum of all values Franked to departments
Items	long	Long	Out	Any	Sum of items Franked to departments

Reads the total value and the number of items Franked to departments.

Statistics (FMStatistic)

Important: Before any of the methods of FMStatistic can be used the property ActiveConnection must be set.

Properties

ActiveConnection – Relation to Connection

`IConnection ActiveConnection`

C++ Type	VB Type	R/W	Value range
IConnection*	Connection	Read-Write	Any

Has to be set before the first use of any method of FMStatistic. It defines the Connection that FMStatistic uses to perform the functions in the FM.

Methods

DailyTotal – Read daily statistics

`DailyTotal(Value, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Value	CURRENCY	Currency	Out	any	Value Franked on current day
Items	long	Long	Out	any	Number of items Franked on current day

Reads value and number of items Franked on current day.

MonthlyTotal – Read 'Monthly Total' statistics

`MonthlyTotal(Value, Items, Day, Month, Year)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Value	SAFEARRAY(CURRENCY)	Currency()	Out	any	One-dimensional array of values Franked
Items	SAFEARRAY(long)	Long()	Out	any	One-dimensional array of number of items Franked
Day	SAFEARRAY(short)	Integer()	Out	1...31	One-dimensional array of Days
Month	SAFEARRAY(short)	Integer()	Out	1..12	One-dimensional array of Months
Year	SAFRARRAY(short)	Integer()	Out	any (1980…2099)	One-dimensional array of Years (4 digit format)

Provides a table containing the values and number of items franked each day. All arrays contain the same number of elements. The elements with the same index belong together and represent the statistics of one day (value, items and date). The entries are sorted by date (oldest first). MonthlyTotal does not include the current day.
With this data a monthly report of daily consumption can be generated.

SinceTotalClear – Read and clear 'Since Total' statistics

`SinceTotalClear(Value, Items, Day, Month, Year)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Value	CURRENCY	Currency	Out	any	Total value Franked since last clear
Items	long	Long	Out	any	Total number of items Franked since last clear
Day	short	Integer	Out	1...31	Day of last clear
Month	short	Integer	Out	1..12	Month of last clear
Year	short	Integer	Out	any (1980…2099)	Year (4 digit format) of last clear

Reads total value and number of items that have been Franked since the last clear of 'Since Total' counter and resets the counters to zero.

SinceTotalRead – Read 'Since Total' statistics

```
SinceTotalRead(Value, Items, Day, Month, Year)
```

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Value	CURRENCY	Currency	Out	any	Total value Franked since last clear
Items	long	Long	Out	any	Total number of items Franked since last clear
Day	short	Integer	Out	1...31	Day of last clear
Month	short	Integer	Out	1..12	Month of last clear
Year	short	Integer	Out	any (1980…2099)	Year (4 digit format) of last clear

Reads total value and number of items that have been Franked since the last clear of 'Since Total' counter.

TariffGroups – Tariff group statistics

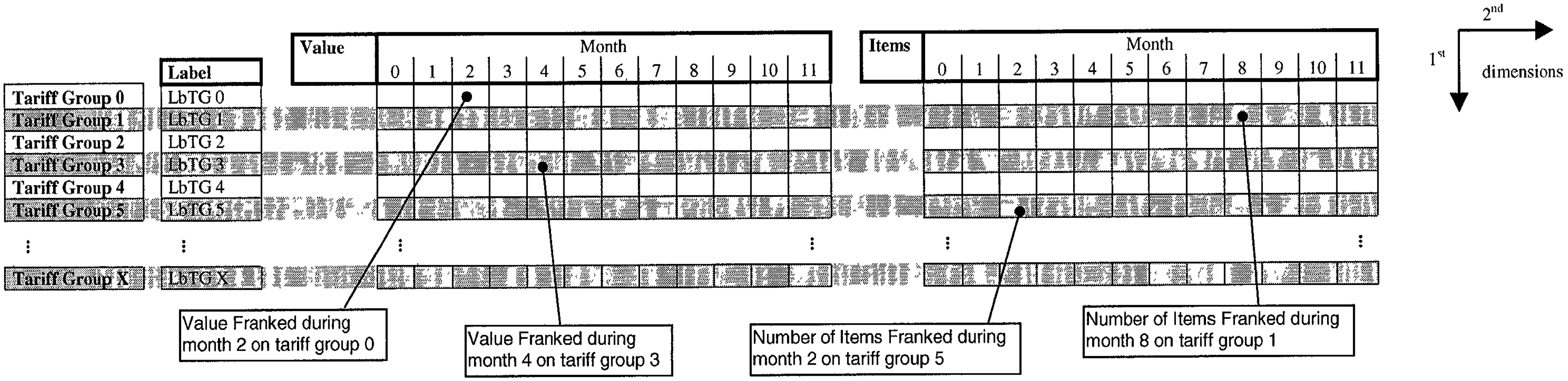
`TariffGroups(Label, Value, Items)`

Parameter	C++ Type	VB Type	I/O/R	Value range	Description
Label	SAFEARRAY(BSTR)	String()	Out	any	One-dimensional array of the labels of the tariff groups (Text of max. 25 chars)
Value	SAFEARRAY(CURRENCY)	Currency()	Out	any	Two-dimensional array of values Franked. 1st dim. = values for different tariff groups 2nd dim. = values for different months
Items	SAFEARRAY(long)	Long()	Out	any	Two-dimensional array of number of items Franked 1st dim. = items for different tariff groups 2nd dim. = items for different months

The tariff group statistics always contain data for the last 12 months.

This function provides a data structure that contains the following for each tariff group: one label, 12 values and 12 numbers of items (12 because there is one for each month).

The data is organized as shown below:



Month 0 is always the current month whereas month 1 to 11 are the previous months. E.g.: on November 15 1999, month 0 is November 1999, month 1 is October 1999 and month 11 is December 1998.

Type Definitions

The following definitions are all enumerated types. Do *not* use the discrete values of the constants, as they might change in later versions of the program library.

BASES

B120	
B150	
B220	
B220P	
B250P	
B320	
B320P	
B335	
B335P	
B335R	Low cost base
B337	
B337P	
B340P	High speed base
BTEST	Special base for duration tests

FMTYPES

F3XXPLUS	FMs of the F3XXPLUS series – the only FM-type that library version 1.0 does work with

FRANKMODES

FRK_NORM	Normal Franking without Preselection. Check property for availability.
FRK_TAPE	Preselected tape Franking. Possible with Bases that have a tape motor only. Check property for availability.
FRK_LETTER	Preselected letter Franking. Available with some Bases only. Check property for availability.
FRK_ITEM	Preselected item Franking. Tapes or letters can be Franked. Check property for availability

See also *FrankModeAvailItem – Availability of Franking Mode FRK_ITEMS* on page 7 as well as the three following properties.

LASTDECTYPES

LD05	Last decade wheel can be 0 or 5 only
LD09	Last decade wheel can be anything from 0...9

See also *LastDecadeType – Type of the last decade wheel* on page 8.

RETVALS

The first letter indicates the type of the return value:

R: Normal return value
E: Error
W: Warning
L: Limit

General

Standard return values.

R_OK	OK, function successful
E_NOK	Not OK, General error. Usually such an error causes an exception.
R_NEWDATE	Print date has changed and must be adjusted ether by hand or by pushing a button on the FM keyboard and the print must be checked. Note: An offset of a few hours between the print date and the FM's system date can be defined in the FM.
R_QUIET	FM in QUIET state, not ready for Franking (anymore).
R_HVLIM	Postage higher than High Value limit. Call HVLimitDeblock or HVLimitAbort within the timeout.
E_TIMEOUT	High Value limit timeout has run down. SetDecades must be repeated.

Warnings

Warning limit reached. A warning indicates that soon a limit will be reached which can cause the FM to be blocked. If a warning is returned from a SetDecade command, the FM is NOT ready for Franking. In that case repeat the SetDecade command .

W_DESCREG	Descending register warning limit reached (soon no more funds).
W_ASCREG	Ascending register warning limit reached.
W_MAXITEMS	Item warning limit reached.
W_READING	Reading warning limit reached. Soon a TMS connection must be established. (Occurs on TMS meters only)
W_BATTERY	Battery expiration date reached.

Limits

Franking is not possible anymore and an action has to be taken to make Franking available again.

L_DESCREG	No more funds (descending register is zero).
L_ASCREG	Ascending register limit reached.
L_MAXITEMS	Item limit reached.
L_READING	Reading limit reached. A TMS connection must be established. (Occurs on TMS meters only)
L_BATCHCOUNTER	Batch counter full. Clear batch counter.
L_DPT	Department counter full. Clear content of department.
L_TOT	Since Total statistics counter full. Clear Since Total statistics.

OnRotorError event

These are return values used by OnRotorError event only.

E_CALLSERVICE	An error that demands to call a service technician.
E_ROTOR	Error on rotor e.g. position.
E_CROSS	Cross error
E_BASESPEED	Speed of base to high.
E_DECADE	Decade wheels can not be set.
E_PRINTER	Printer error, e.g. not ready or not present.

Departments

These return values are used in department functions only.

E_EXIST	Department does exist already. Can not open department.
E_NOTCLEAR	Department must be clear before it can be closed.
E_FULL	No more room for departments. Can not open department.

TMS

These return values are used in OnTMSEnd only.

E_DESC_MAX	Descending register exceeded maximum.
E_SUM_OFL	Control Total or Descending register overflow.
E_AMOUNT_MAX	Amount too high.
E_AMOUNT_STEP	Amount is not a multiple of the defined step. Round amount.
E_AMOUNT_MIN	Amount too low.
E_TMS_ABORT	TMS host has aborted.
E_TMS_TRANSM	Error in TMS transmission.
E_TMS_CONNECT	No TMS connection.
E_TMS_MODEM	TMS modem error.
E_TMS_LINE_BUSY	Phone line is busy.
E_TMS_MODEM_INIT	Error initializing modem.
E_TMS_FAILED	TMS failed (internal problem or not a TMS FM).
E_FMSTATE	FM is in wrong state for carrying out this function. Go into QUIET state first.

HSB (High Speed Base – S340 plus)

These return values indicate the state of the HSB.

E_HSB_KEY	Key of high speed base is in wrong position.
E_HSB_HOT	High speed base too hot.
E_HSB_STANDBY	High speed base is on standby.
E_HSB_INKCOVER	Ink cover of high speed base is open.

Exceptions (EXCEPTIONIDS)

The exceptions thrown by the FM Control library can be identified by the ID listed below. (Hint: In Visual Basic the exception IDs returned by the Err object do have an offset of the value of vbObjectError. Use vbObjectError to calculate the ID before comparing Err.Number with the IDs.)

ID	**Description** (also returned as Description in the Err object)
NOCONNECTION	Connection could not be established
INVALIDACTCON	No connection (invalid ActiveConnection)
INVCONSTRING	Invalid Connection String
CONEXIST	Connection exists already
LOSTCONNECTION	Connection lost
NOANSWER	FM does not answer
SENDFAIL	Send failed
INVSENDDATA	Invalid data to send to FM
INCORRDATA	FM said 'data is incorrect'
INVFMDATA	Invalid data received from FM
INVPOSTAGE	Invalid Postage
INVDPT	Invalid DPT number (negative or too many digits)
DPTNEXIST	DPT does not exist
INVMAILCLASS	Invalid Mail Class
INVFRKMODE	Invalid Franking Mode
INVPRESELECTION	Invalid Preselection value
INVMARGIN	Invalid Margin value
WRONGSTATE	FM is in wrong state for carrying out this function
STANDBY	Rotor task is on standby
DCOPEN	Die Cover is open
FCTNOTSUPPORTED	Function not supported for this FM type
ABORT	Current action aborted
NOK	FM returned NOT OK
UNKNOWN	Unknown exception

Requirements

PC system requirements

Operating system: Windows95 or higher or WindowsNT 4.0 with Service Pack 4 or higher.

Software: MSVCRT.DLL version 6.0 or higher. See below for instructions for checking version and updating. Distributed COM (DCOM) installed. This is included in WindowsNT 4.0 and Windows98 or higher. If you are using Windows95 and are not sure whether DCOM is installed on your system please install DCOM by starting DCOM95.EXE that comes together with FMCtrl.

Hardware: At least one free serial COM-Port (V.24 / RS232) – one per FM to be controlled simultaneously. A SCSI Terminal Server can be used to expand the PC with up to 32 additional COM-Ports. Such SCSI Terminal Servers are available from Digi (www.digi.com).
V.24 / RS232 link cable to connect the PC with the FM.

FM requirements

FM-Type: F3XXPLUS with one free V.24 / RS232 port.

FMs using an external modem for TMS can be equipped with an INFAC interface board. INFAC provides two V.24 / RS232 ports. One port to be used for the connection to the PC, the other for the optional external modem.
Remark: While in FM Control mode (Remote Control = ON) no other devices than the PC or the external modem can be connected to the FM. Additional periphery like scales would have to be interfaced to the PC.

FM-Software: Remote control enabled software. FMs having such a software installed do have the option to switch remote control ON or OFF in the Service menu.

Before using the program library

Version 6.0 of MSVCRT.DLL

MSVCRT.DLL is an important part of the WindowsNT system and is often used in Windows95 too. To make sure that the FMCtrl library works correctly, version 6.0 or higher of MSVCRT.DLL is necessary. So before the first use of the FMCtrl library check the existence and version of MSVCRT.DLL and update it if necessary. To do this proceed as follows:

- Use Windows Explorer to open the directory where MSVCRT.DLL is located (WindowsNT: `C:\WinNT\System32` Windows95: `C:\Windows\System`)

If MSVCRT.DLL does not exist:

- Copy the file delivered together with the FMCtrl library to the location mentioned above.

If MSVCRT.DLL does exist:

- Select MSVCRT.DLL and choose Properties form File menu.
- In the appearing window select Version. On the first line the version number is shown.

If the existing MSVCRT.DLL is older than version 6.0:

- Open a DOS console window or switch into DOS mode.
- Change to directory where MSVCRT.DLL is located (see above).
- Rename MSVCRT.DLL to MSVCRT.OLD by entering `rename msvcrt.dll msvcrt.old` (this does work from the DOS console window or DOS mode only)
- Copy MSVCRT.DLL that comes together with FMCtrl to the directory where you found MSVCRT.DLL.
- Reboot your system

Registering the library

Before the FMCtrl library can be used it must be registered in Windows:

- Open a DOS console window.
- Change to directory where FMCtrl.exe is located (this can be any directory).
- Enter the following commands (the first command unregisters a possibly registered older version of the library – always do this before registering):

```
FMCtrl-unregserver
FMCtrl-regserver
```

- Now FMCtrl is ready for use e.g. from Visual Basic. To use it from Visual Basic 6.0 open a new project, choose References from Project menu and tick the FMCtrl entry. Open the object browser by pressing F2. In the object browser select FMCTRLLib from the drop down list. Now the interface of the library is shown in the object browser.

Using the library

Establishing connection

Before any of the functions in the library can be used a Connection object has to be created. The next step is usually calling the Connect method that establishes the connection to the FM.

Each of the other objects has an ActiveConnection property. It defines the Connection that the object uses to perform the functions in the FM. The ActiveConnection must be assigned with a Connection before any property or method of the object can be used.

Error handling

Most of the functions in the FM Control library can throw COM-Exceptions in case of an error. COM-Exceptions provide an error number and a description, that can be used in the error handler of the user application.

Some functions do also return a value that determinates the state of the machine. Those values do indicate the reaching of warning levels and limits rather than errors.

Logfile

If you encounter problems using the FM Control library the logfile can provide additional information about the failure. Logfile writing can be enabled by creating a subdirectory named `Log` in the directory where FMCtrl.exe is located. The logfile `fmctrl.txt` will be placed into this subdirectory.
Please always create a logfile before contacting technical support!

Connection (Connection)

Properties

ConnectionString – Connection string

`BSTR ConnectionString`

C++ Type	VB Type	R/W	Value range	Description
BSTR	String	Read-Only	valid strings	Defines connection device, protocol etc.

Returns the ConnectionString for the current connection. All characters of this string are always upper case. If the string is not empty (length ≠ 0) the connection does exist. On disconnection the string is reset to 0 length.

LibraryVersion – Version number

`BSTR LibraryVersion`

C++ Type	VB Type	R/W	Value range	Description
BSTR	String	Read-Only	#.#.#.# where # is a number 0...9999	Version of this program library

Returns the version of the program library that is used.

States with reactions

The following list contains all states returned from the Franking Machine with a description and the reaction that is taken by the FM Control library. The FM Control library ether throws an exception or returns a certain value to inform the user application.

State	Description	Reaction (return value [RETVAL] or exception)
ST_OK	Function successful	R_OK
ER_NOK	Error	E_NOK → Exception: "FM sent NOT OK" (not for Events)
ST_QUIET	Meter not ready (e.g. due to rejection of „high value")	R_QUIET
ST_NEW_DATE	Print date has changed	R_NEWDATE
ST_Wr_DptNr	Wrong department number	Exception: "Invalid DPT number (e.g. too many digits)"
ST_Dpt_Nexist	Department number does not exist	Exception: "DPT does not exist"
ST_Dpt_Exist	Department exists already (when trying to open DPT)	E_EXIST
ST_Ndpt_open	No Department open (when reading or clearing total)	Exception: "DPT does not exist"
ST_NoDptSpace	Do space for more departments	E_FULL
ST_ClrContent	Department is not clear (when trying to close a DPT)	E_NOTCLEAR
ST_EoList	remapped to ST_OK	remapped to ST_OK
ER_PARAMETER	Parameter value invalid	Exception: "FM said 'data is incorrect'"
ER_DC_OPEN	Please close die cover	Exception: "Die Cover is open"
ER_FORMAT	Invalid value or invalid mail class or invalid preselection mode (neither FRK_NORM nor FRK_ITEM)	Exception: "FM said 'data is incorrect'"
ER_CALL_SERVICE	CS appeared or exists already	E_CALLSERVICE (also used with ROTOR-ERROR event)
ER_BREAK	Current action aborted / Setting error	Exception: "Current action aborted"
ER_HV_TIMEOUT	HV timer run down --> meter not ready to frank	E_TIMEOUT
ER_STANDBY	Please repeat procedure	Exception: "Rotor task is on standby"
ER_ROTOR	Rotor not in base position	E_ROTOR (also used with ROTOR-ERROR event)
ER_CROSS	Cross error	E_CROSS (also used with ROTOR-ERROR event)
ER_BASE_SPEED	Speed of Base to high	E_BASESPEED (also used with ROTOR-ERROR event)
ER_DECADE	Decade wheels can not be set	E_DECADE (also used with ROTOR-ERROR event)
ER_MARGIN_LOW	Margin too low	Exception: "Invalid Margin value"
ER_MARGIN_HIGH	Margin too high	Exception: "Invalid Margin value"
ER_DATLEN	Number of bytes (parameter length) incorrect	Exception: "FM said 'data is incorrect'"
ER_NOPRESEL	No preselection entered	Exception: "Invalid Preselection value"
ER_INVPRESEL	Wrong preselection (0) entered	Exception: "Invalid Preselection value"
ER_PRINTER_NOT_READY	Journal printer not ready (no paper, no connection etc.)	E_PRINTER (also used with ROTOR-ERROR event)
ER_MSTATE	FM in wrong state for carrying out requested function	Exception: "FM is in wrong state for carrying out this function" Special: E_FMSTATE for OnTMSEnd event.
L_HV	Postage above HV limit (state returned with I-Message)	R_HVLIM
L_ASC_REG	Ascending register limit reached	L_ASCREG
L_DESC_REG	No more funds (descending register is zero)	L_DESCREG

State	Description	Reaction (return value [RETVAL] or exception)
L_MAX_ITEMS	Item limit reached	L_MAXITEMS
L_VAL_READING	Reading limit (value) reached	L_READNING
L_ITEM_READING	Reading limit (items) reached	L_READNING
L_TIME_READING	Reading limit (time) reached	L_READNING
L_DPT_ITEM	Clear content if this department	L_DPT
L_DPT_VALUE	Clear content if this department	L_DPT
L_TOT_ITEM	Clear statistics total	L_TOT
L_TOT_VALUE	Clear statistics total	L_TOT
L_BAT_ITEM	Clear batch total	L_BATCHCOUNTER
L_BAT_VALUE	Clear batch total	L_BATCHCOUNTER
W_ASC_REG	Warning: ascending register limit **ALL WARNINGS: Decades are NOT set → repeat Decade setting**	W_ASCREG
W_DESC_REG	Warning: descending register limit (soon no more funds)	W_DESCREG
W_MAX_ITEMS	Warning: item limit	W_MAXITEMS
W_VAL_READING	Warning: Reading limit (value)	W_READING
W_ITEM_READING	Warning: Reading limit (items)	W_READING
W_TIME_READING	Warning: Reading limit (time)	W_READING
W_BATTERY	Warning: battery expiration date	W_BATTERY
ER_DESC_MAX	Max. descending register value exceeded	E_DESC_MAX
ER_SUM_OFL	Control total or descending register overflow	E_SUM_OFL
ER_VORG_MAX	(TMS) Amount too high	E_AMOUNT_MAX
ER_VORG_PAR	(TMS) Amount is not a multiple of the defined step	E_AMOUNT_STEP
ER_VORG_MIN	(TMS) Amount too low	E_AMOUNT_MIN
ER_USER_ABORT	TMS reset aborted by the user	*Not implemented (yet)*
ER_TMS_ABORT	TMS host has aborted	E_TMS_ABORT
ER_TMS_ABORT_MSG	TMS host has aborted (with message)	E_TMS_ABORT
ER_TMS_TRANSM	Error in TMS transmission	E_TMS_TRANSM
ER_TMS_CONNECT	No TMS connection	E_TMS_CONNECT
ER_TMS_MODEM	TMS modem error	E_TMS_MODEM
ER_TMS_MODEM_INIT	(TMS) Error initializing modem	E_TMS_MODEM_INIT
ER_TMS_PHONE_BUSY	(TMS) Line is busy	E_TMS_LINE_BUSY
ER_TMS_FAILED	TMS failed (internal problem or not a TMS FM)	E_TMS_FAILED
ER_HSB_MS	Key of high speed base is in wrong position	E_HSB_KEY
ER_HSB_HOT	High speed base too hot	E_HSB_HOT
ER_HSB_STBY	High speed base is on standby	E_HSB_STANDBY
ER_HSB_FD	Ink cover of high speed base is open	E_HSB_INKCOVER

Return value to a O_ message which is sent while the FM is in the wrong STATE, the answer is an R_ message with ER_MSTATE (not I_CANCEL).

List of FM functions with returned states

ID	Function name	States
1E72	R_SPC_CONNECT	ST_OK, ER_NOK
1E13	R_ASK_BASE_MODEL_PC	ST_OK, ER_NOK
1E15	R_ASK_BASE_SW_VERS_PC	ST_OK, ER_NOK
1E76	R_ROTPARAM_READ	ST_OK, ER_NOK
1E3E	R_READ_DATETIME	ST_OK, ER_NOK
1E17	R_USER_TIMEOUT_ONOFF_PC	ST_OK, ER_NOK
1E19	R_LCW_ONOFF_PC	ST_OK, ER_NOK
1E22	R_READ_MACH_NR	ST_OK, ER_NOK
1E97	R_READ_NO_MACH	ST_OK, ER_NOK
1E9B	R_READ_NO_REMPL	ST_OK, ER_NOK
1E32	R_HV_ONOFF_PC	ST_OK, ER_NOK
1E34	R_DISP_DESC_ONOFF_PC	ST_OK, ER_NOK
1E01	R_KEYBOARD_ONOFF_PC	ST_OK, ER_NOK
1EBD	R_BLK_FRANK_MENU_PC	ST_OK, ER_NOK
1E36	R_BAT_ON_PC	ST_OK, ER_NOK
1E38	R_BAT_READ_PC	ST_OK, ER_NOK
1E3A	R_BAT_CLEAR_PC	ST_OK, ER_NOK
1E7C	R_GET_COUNTER	ST_OK, ER_NOK
1ECC	R_PC_READ_MC_TEXT	ST_OK, ER_NOK, ?ER_MSTATE
1E11	R_TEXT_TO_DISPLAY_PC	ST_OK, ER_NOK, ER_DATLEN, ER_FORMAT
1E6A	R_SPC_MC_COR	ST_OK, ER_STANDBY, ER_CALL_SERVICE, ER_DC_OPEN, ER_DATLEN, ER_PARAMETER
1E63	R_SPC_GO_QUIET	ST_OK, ER_ROTOR, ER_DC_OPEN, ER_CALL_SERVICE
1E74	R_SPC_DEC_SET	All*
1EBF	R_PC_DEC_SET_TAX	All*
1E68	I_SPC_QUIET	ST_OK, ER_HSB_MS, ER_HSB_HOT, ER_HSB_STBY, ER_HSB_FD
1E5D	I_PC_FRK_EVENT	ST_OK, ST_QUIET, ST_NEW_DATE?, W_..., L_...
1E4D	I_SPC_ROT_ERROR	ER_CALL_SERVICE, ER_ROTOR, ER_CROSS, ER_BASE_SPEED, ER_DECADE, ER_PRINTER_NOTREADY
1E03	R_SPC_DISCONNECT	[redacted] ST_OK, ER_ROTOR, ER_DC_OPEN, ER_CALL_SERVICE
1E3C	R_SET_TMS_PC	ST_OK, ER_TMS_ABORT, ER_TMS_ABORT_MSG, ER_TMS_TRANSM, ER_TMS_CONNECT, ER_TMS_MODEM, ER_TMS_PHONE_BUSY, ER_TMS_MODEM_INIT, ER_TMS_FAILED
1EA7	R_MailClass_PC	ST_OK, ?ER_DATLEN
1EA5	R_TOT_READ_PC	ST_OK, ?ER_DATLEN
1EBB	R_TOT_CLEAR_PC	ST_OK, ?ER_DATLEN
1EA9	R_READ_DAY_PC	ST_OK, ?ER_DATLEN
1EAF	R_READ_YEAR_PC	ST_OK, ?ER_DATLEN
1EB7	R_READ_Tgroup_Par	ST_OK, ?ER_DATLEN
1EA1	R_1ST_DPT_PC	ST_OK, ?ER_DATLEN, ST_EoList
1EA3	R_NEXT_DPT_PC	ST_OK, ?ER_DATLEN, ST_EoList
1EAB	R_1ST_DAY_PC	ST_OK, ?ER_DATLEN, ST_EoList
1EAD	R_NEXT_DAY_PC	ST_OK, ?ER_DATLEN, ST_EoList
1EB1	R_1ST_TarifGroup_PC	ST_OK, ?ER_DATLEN, ST_EoList
1EB3	R_NEXT_TarifGroup_PC	ST_OK, ?ER_DATLEN, ST_EoList
1E6C	R_DPTPARAM_READ	?
1E0D	R_DPT_OPEN_PC	ST_OK, ST_Wr_DptNr, ST_Dpt_Exist , ST_NoDptSpace
1E0F	R_DPT_CLOSE_PC	ST_OK, ST_Wr_DptNr, ST_Dpt_Nexist , ST_ClrContent
1E53	R_GET_DPT_PC	ST_OK, ST_Wr_DptNr, ST_Dpt_Nexist
1E55	R_CLEAR_DPT_PC	ST_OK, ST_Wr_DptNr, ST_Dpt_Nexist
1E57	R_SUM_READ_PC	ST_OK, ST_Ndpt_open
1E59	R_SUM_CLEAR_PC	ST_OK, ST_Ndpt_open

APPENDIX II

```
Attribute VB_Name = "Conversions"
'Module Conversions contains routines to convert values form one
'type to another.

Option Explicit

'Converts enumerated type FMTYPES to a string with the name

Function FMTypeToString(FMType As FMTYPES) As String
    Select Case FMType
    Case F3XXPLUS
        FMTypeToString = "F3XXPLUS"
    Case Else
        FMTypeToString = "(unknown)"
    End Select
End Function

'Converts enumerated type BASES to a string with the name

Function BaseModelToString(Base As BASES) As String
    Select Case Base
    Case B120
        BaseModelToString = "B120"
    Case B150
        BaseModelToString = "B150"
    Case B220
        BaseModelToString = "B220"
    Case B220P
        BaseModelToString = "B220P"
    Case B250P
        BaseModelToString = "B250P"
    Case B320P
        BaseModelToString = "B320P"
    Case B335
        BaseModelToString = "B335"
    Case B335P
        BaseModelToString = "B335P"
    Case B335R
        BaseModelToString = "B335R"
    Case B337
        BaseModelToString = "B337"
    Case B337P
        BaseModelToString = "B337P"
    Case B340P
        BaseModelToString = "B340P"
    Case BTEST
        BaseModelToString = "BTEST"
    Case Else
```

```
        BaseModelToString = "(unknown)"
    End Select

End Function

'Composes the print image out of the information read from the FM
and
'returns it as string (e.g. 99.99 or 9999900 etc.)

Function CreatePrintImage() As String
    Dim Config As New FMCTRLLib.FMConfig        'Create an new FMC
onfig object
    Dim strPrintImage As String                 'Define needed var
iables
    Dim strFixZeros As String
    Dim strPostPoint As String
    Dim strPrePoint As String

    '!!! Error handling must be done in the calling function !!!

    Config.ActiveConnection = fMainForm.Con     'Define the connec
tion to be used by Config

    strPrintImage = String(Config.DecadeNumber - 1, "9")    'A "9"
 for each decade wheel do not add the last wheel yet
    If Config.LastDecadeType = LD09 Then
        strPrintImage = strPrintImage + "9"     'Add a "9" for the
 last decade wheel if last decade can be 0..9
    Else
        strPrintImage = strPrintImage + "5"     'Add a "5" for the
 last decade wheel if last decade be only 0 or 5
    End If

    strFixZeros = String(Config.FixedZeros, "0")    'A "0" for eac
h fixed zero
    strPrintImage = strPrintImage + strFixZeros     'Add the zeros
 to the decade wheels

    If Config.DecPointPosition > 1 Then             'If FM uses a
decimal point
        strPostPoint = Right(strPrintImage, Config.DecPointPositio
n - 1)    'Copy the characters that follow the decimal point
        strPrePoint = Left(strPrintImage, Len(strPrintImage) - Con
fig.DecPointPosition + 1) 'Copy the characters that are before the
 .
        strPrintImage = strPrePoint + "." + strPostPoint    'Inser
t the decimal point
    End If
    CreatePrintImage = strPrintImage            'Return composed s
tring

    Set Config = Nothing                        'Disassociate Conf
ig object
End Function

'Creates a format mask that can be used to format money values for
 displaying.
'Attention: the "," (thousand sepparator) and the "." (decimal poi
nt) will be
'            automatically replaced by the signs defined in the Win
dows country
'            settings (e.g. "'" and ",").

Function CreateCurrencyFormatMask() As String
    Dim Config As New FMCTRLLib.FMConfig        'Create an new FMC
onfig object
    Dim strMask As String                       'Define needed var
iables

    '!!! Error handling must be done in the calling function !!!

    Config.ActiveConnection = fMainForm.Con     'Define the connec
tion to be used by Config

    strMask = "##,##0"                          'Define the initia
l mask
    If Config.DecPointPosition > 1 Then         'If FM uses a deci
mal point
                                                'Add a "." and as
many "0" as there are decades in the FM after the point
        strMask = strMask + "." + String(Config.DecPointPosition -
 1, "0")
    End If

    CreateCurrencyFormatMask = strMask          'Return mask
    Set Config = Nothing                        'Disassociate Conf
ig object
End Function

'Converts a True to a "Yes" and a False to a "No"

Function BoolToYesNo(bBool As Boolean) As String
    If bBool Then
        BoolToYesNo = "Yes"
    Else
```

```
        BoolToYesNo = "No"
    End If
End Function

'Composes a string containing a list of the names of all available
 Franking Modes

Public Function FrankModesToString() As String
    Dim Config As New FMCTRLLib.FMConfig           'Create an new FMC
onfig object
    Dim strFrkModes As String                      'Define needed var
iables

    '!!! Error handling must be done in the calling function !!!

    Config.ActiveConnection = fMainForm.Con        'Define the connec
tion to be used by Config
    strFrkModes = ""

    If Config.FrankModeAvailNorm Then              'If FRK_NORM is av
ailable
        strFrkModes = "Normal"                     'Set string
    End If
    If Config.FrankModeAvailTape Then              'If FRK_TAPE is av
ailable
        If Len(strFrkModes) > 0 Then               'If there is alrea
dy an entry
            strFrkModes = strFrkModes + " / "      'Insert a /
        End If
        strFrkModes = strFrkModes + "Tapes"        'Add Tapes
    End If
    If Config.FrankModeAvailLetter Then            '..
        If Len(strFrkModes) > 0 Then
            strFrkModes = strFrkModes + " / "
        End If
        strFrkModes = strFrkModes + "Letters"
    End If
    If Config.FrankModeAvailItem Then              '..
        If Len(strFrkModes) > 0 Then
            strFrkModes = strFrkModes + " / "
        End If
        strFrkModes = strFrkModes + "Items"
    End If

    FrankModesToString = strFrkModes               'Return string

    Set Config = Nothing                           'Disassociate Conf
ig object
```

```
End Function

'Converts an index (used to identify the Franking Mode in a ComboB
ox.ItemData)
'to enumerated type FRANKMODES

Public Function IndexToFrankMode(Index As Integer) As FRANKMODES
    Select Case Index
    Case 0
        IndexToFrankMode = FRK_NORM
    Case 1
        IndexToFrankMode = FRK_TAPE
    Case 2
        IndexToFrankMode = FRK_LETTER
    Case 3
        IndexToFrankMode = FRK_ITEM
    End Select
End Function

'Select the whole text of the given TextBox.

Public Sub SelectAll(txtBox As TextBox)
    txtBox.SelStart = 0
    txtBox.SelLength = Len(txtBox.Text)
End Sub

'Get the appropriate text message to a return value from the FM Co
ntrol library.

Public Function RetValToText(RetVal As RETVALS) As String
    Dim s As String

    Select Case RetVal
    Case R_OK
        s = "OK, function successful."
    Case E_NOK
        s = "Not OK, general error."
    Case R_NEWDATE
        s = "Print date has changed."
    Case R_QUIET
        s = "FM in QUIET state."
    Case R_HVLIM
        s = "Postage is above High Value limit."
    Case E_TIMEOUT
        s = "Timeout has run down."
    Case E_CALLSERVICE
        s = "Call a service technician."
    Case E_ROTOR
```

```
        s = "Error on rotor."
    Case E_CROSS
        s = "Cross error."
    Case E_BASESPEED
        s = "Speed of base to high."
    Case E_DECADE
        s = "Decade wheels can not be set."
    Case E_PRINTER
        s = "Printer error."
    Case W_DESCREG
        s = "Low Credit warning (soon no more funds)."
    Case W_ASCREG
        s = "Ascending counter warning level reached."
    Case W_MAXITEMS
        s = "Item counter warning level reached."
    Case W_READING
        s = "Reading warning level reached."
    Case W_BATTERY
        s = "Battery expiration date reached."
    Case L_DESCREG
        s = "No more funds (descending counter is zero)."
    Case L_ASCREG
        s = "Ascending register limit reached."
    Case L_MAXITEMS
        s = "Item counter limit reached."
    Case L_READING
        s = "Reading limit reached."
    Case L_BATCHCOUNTER
        s = "Batch counter full."
    Case L_DPT
        s = "Department counter full."
    Case L_TOT
        s = "Since Total statistics counter full."
    Case E_HSB_KEY                                  'RetVals for H
SB (high speed base)
        s = "Key of high speed base is in wrong position."
    Case E_HSB_HOT
        s = "High speed base too hot."
    Case E_HSB_STANDBY
        s = "High speed base is on standby."
    Case E_HSB_INKCOVER
        s = "Ink cover of high speed base is open."
    Case Else
        s = "Error: unhandled RetVal!"
    End Select

    RetValToText = s
End Function
```

```
'Get the appropriate short text message to a return value from the
 FM Control library.
'These short texts are used for displaying in the Frankings list o
f frmFranking

Public Function RetValToShortText(RetVal As RETVALS) As String
    Dim s As String

    Select Case RetVal
    Case R_OK
        s = "OK"
    Case E_NOK
        s = "Error"
    Case R_NEWDATE
        s = "New Date"
    Case R_QUIET
        s = "QUIET"
    Case R_HVLIM
        s = "HV limit"
    Case E_TIMEOUT
        s = "Timeout"
    Case E_CALLSERVICE
        s = "Call service"
    Case E_ROTOR
        s = "Rotor error"
    Case E_CROSS
        s = "Cross error"
    Case E_BASESPEED
        s = "Base speed"
    Case E_DECADE
        s = "Decade error"
    Case E_PRINTER
        s = "Printer error"
    Case W_DESCREG
        s = "Low Credit"
    Case W_ASCREG
        s = "Warning: Ascending"
    Case W_MAXITEMS
        s = "Warning: Items"
    Case W_READING
        s = "Warning: Reading"
    Case W_BATTERY
        s = "Warning: Battery"
    Case L_DESCREG
        s = "No more funds"
    Case L_ASCREG
        s = "Limit: Ascending counter"
    Case L_MAXITEMS
        s = "Limit: Item counter"
    Case L_READING
        s = "Limit: Reading"
    Case L_BATCHCOUNTER
        s = "Batch counter full"
    Case L_DPT
        s = "Department full"
    Case L_TOT
        s = "Statistics full"
    Case Else
        s = "Error: unhandled RetVal!"
    End Select

    RetValToShortText = s
End Function

'Displays the appropriate message box to a return value from the F
M Control library.
'Some of the messages need to be answered and an action must be ta
ken.

Public Sub ProcessRetVal(RetVal As RETVALS)
    Dim msg As String
    Dim MsgBoxStyle As VbMsgBoxStyle
    Dim bShowMsgBox As Boolean
    Dim NL As String

    NL = Chr(13)                        'Define newline character
    bShowMsgBox = True
    msg = RetValToText(RetVal)          'Get the text message for
the RetVal

    Select Case RetVal
    Case R_OK
        bShowMsgBox = False             'Do not show message box
        frmFranking.sbStatusBar.SimpleText = "Ready for Franking"
  'Write to StatusBar of frmFranking
    Case E_NOK
        MsgBoxStyle = vbCritical
    Case R_NEWDATE
        msg = msg + NL + "Confirm new date?"
        MsgBoxStyle = vbYesNo + vbQuestion          'Message box w
ith Yes and No buttons
        If vbYes = MsgBox(msg, MsgBoxStyle) Then    'If Yes clicke
d
            ConfirmNewDate                          'Confirmation
is sent to FM
```

```
        End If
        bShowMsgBox = False                             'Do not show a
dditional message box
    Case R_QUIET
        frmFranking.sbStatusBar.SimpleText = "Quiet"    'Write to
StatusBar of frmFranking
        MsgBoxStyle = vbInformation
        bShowMsgBox = False                             'Do not show m
essage box
    Case R_HVLIM
        msg = msg + NL + "Accept high postage?"
        MsgBoxStyle = vbYesNo + vbQuestion              'Message box w
ith Yes and No buttons
        AnswerHVLimit (MsgBox(msg, MsgBoxStyle))        'Send message
to FM depending on the button pressed
        bShowMsgBox = False                             'Do not show a
dditional message box
    Case E_TIMEOUT
        msg = msg + NL + "Decades not set."
        MsgBoxStyle = vbInformation
    Case E_CALLSERVICE, E_ROTOR, E_CROSS, E_BASESPEED, E_DECADE, E
_PRINTER
        msg = msg + NL + "Decades not set."
        MsgBoxStyle = vbCritical
    Case W_DESCREG, W_ASCREG, W_MAXITEMS, W_READING, W_BATTERY
        msg = msg + NL + "Decades not set."
        MsgBoxStyle = vbInformation
    Case L_DESCREG, L_ASCREG, L_MAXITEMS, L_READING, L_BATCHCOUNTE
R, L_DPT, L_TOT
        msg = msg + NL + "Decades not set."
        MsgBoxStyle = vbExclamation
    Case E_HSB_KEY, E_HSB_HOT, E_HSB_STANDBY, E_HSB_INKCOVER
        msg = msg + NL + "Decades not set."
        MsgBoxStyle = vbInformation
    Case Else
        MsgBoxStyle = vbCritical
    End Select

    If bShowMsgBox Then
        MsgBox msg, MsgBoxStyle                         'Show message
box with parameters declared above
    End If
End Sub

'Send HVDeblock or HVAbort depending on the message box result

Private Sub AnswerHVLimit(MsgBoxResult As VbMsgBoxResult)
    Dim Actions As New FMCTRLLib.FMActions      'Create an new FMA
```

```
ctions object
    Dim Ret As RETVALS                          'Define needed var
iables

    '!!! Error handling must be done in the calling function !!!

    Actions.ActiveConnection = fMainForm.Con    'Define the connec
tion to be used by Actions

    If MsgBoxResult = vbYes Then                'If Yes clicked in
 message box
        Ret = Actions.HVLimitDeblock            'Send High Value d
eblock message
    Else                                        'No clicked in mes
sage box
        Ret = Actions.HVLimitAbort              'Send High Value a
bort message
    End If
    Set Actions = Nothing                       'Disassociate Acti
ons object
    ProcessRetVal Ret                           'Process Ret (show
 message if necessary)
End Sub

'Sends New Date Confirmation to FM (the FM then adjusts date if it
 has automatic date, or
'supposes that the user has adjusted the date manually.

Private Sub ConfirmNewDate()
    Dim Actions As New FMCTRLLib.FMActions      'Create an new FMA
ctions object

    '!!! Error handling must be done in the calling function !!!

    Actions.ActiveConnection = fMainForm.Con    'Define the connec
tion to be used by Actions
    Actions.NewDateConfirm                      'Send confirmation
    Set Actions = Nothing                       'Disassociate Acti
ons object
End Sub
```

```
Attribute VB_Name = "ErrorHnd"
Option Explicit

'Displays an error message for the last error occured.
'The exceptions of the FM Control library are converted using vbOb
jectError as offset.

Sub ErrorHandler()
    Dim msg As String
    Dim MsgBoxStyle As VbMsgBoxStyle

    MsgBoxStyle = vbCritical

    Select Case Err.Number
    Case NOCONNECTION + vbObjectError To UNKNOWN + vbObjectError
                'All FM Control ExceptionIDs
        msg = Err.Description + " (" + CStr(Err.Number - vbObjectE
rror) + ")."  'Convert error numbers using vbOjectError
    Case 13           'Type missmatch - occurs when trying to conver
t a text into a number
        msg = "One or more of the entered values are invalid" + "
(" + CStr(Err.Number) + ")."

    Case Else         'Display all other messages as they are
        msg = Err.Description + " (" + CStr(Err.Number) + ")."

    End Select

    'msg = msg + Chr(13) + "Source: " + Err.Source       'Display
source on a new line

    MsgBox msg, MsgBoxStyle                              'Show mess
age
End Sub
```

```
Attribute VB_Name = "Start"
'Application starts with this module

Option Explicit

Public fMainForm As frmMain
Global nComPort As Integer

'Application starts with this Sub

Sub Main()
    Set fMainForm = New frmMain     'Create new frmMain
    fMainForm.Show
End Sub
```

```
VERSION 5.00
Begin VB.Form frmAbout
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "About"
   ClientHeight    =   3630
   ClientLeft      =   45
   ClientTop       =   330
   ClientWidth     =   5865
   ClipControls    =   0   'False
   Icon            =   "frmAbout.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   3630
   ScaleWidth      =   5865
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Tag             =   "About Project1"
   Begin VB.PictureBox picIcon
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      ClipControls    =   0   'False
      Height          =   540
      Left            =   240
      Picture         =   "frmAbout.frx":0442
      ScaleHeight     =   480
      ScaleMode       =   0  'User
      ScaleWidth      =   480
      TabIndex        =   2
      TabStop         =   0   'False
      Top             =   240
      Width           =   540
   End
   Begin VB.CommandButton cmdOK
      Cancel          =   -1  'True
      Caption         =   "OK"
      Default         =   -1  'True
      Height          =   345
      Left            =   4245
      TabIndex        =   0
      Tag             =   "OK"
      Top             =   2625
      Width           =   1467
   End
   Begin VB.CommandButton cmdSysInfo
      Caption         =   "&System Info..."
      Height          =   345
      Left            =   4260
      TabIndex        =   1
      Tag             =   "&System Info..."
      Top             =   3075
      Width           =   1452
   End
   Begin VB.Label lbLibraryVerison
      Caption         =   "Library Version"
      Height          =   255
      Left            =   1050
      TabIndex        =   8
      Top             =   960
      Width           =   4095
   End
```

```
Begin VB.Label Label1
   Caption         =   "Ascom Autelca AG Mailing Systems"
   Height          =   255
   Left            =   1050
   TabIndex        =   7
   Top             =   1320
   Width           =   4095
End
Begin VB.Label lblDescription
   Caption         =   $"frmAbout.frx":0884
   ForeColor       =   &H00000000&
   Height          =   690
   Left            =   1050
   TabIndex        =   6
   Tag             =   "App Description"
   Top             =   1680
   Width           =   4095
End
Begin VB.Label lblTitle
   Caption         =   "Application Title"
   ForeColor       =   &H00000000&
   Height          =   240
   Left            =   1050
   TabIndex        =   5
   Tag             =   "Application Title"
   Top             =   240
   Width           =   4095
End
Begin VB.Line Line1
   BorderColor     =   &H00808080&
   BorderStyle     =   6  'Inside Solid
   Index           =   1
   X1              =   225
   X2              =   5657
   Y1              =   2430
   Y2              =   2430
End
Begin VB.Line Line1
   BorderColor     =   &H00FFFFFF&
   BorderWidth     =   2
   Index           =   0
   X1              =   240
   X2              =   5657
   Y1              =   2445
   Y2              =   2445
End
Begin VB.Label lblVersion
   Caption         =   "Version"
   Height          =   255
   Left            =   1050
   TabIndex        =   4
   Tag             =   "Version"
   Top             =   600
   Width           =   4095
End
Begin VB.Label lblDisclaimer
   ForeColor       =   &H00000000&
   Height          =   825
   Left            =   255
   TabIndex        =   3
   Tag             =   "Warning: ..."
```

```
      Top             =   2625
      Width           =   3750
   End
End
Attribute VB_Name = "frmAbout"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

' Reg Key Security Options...
Const KEY_ALL_ACCESS = &H2003F


' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1                         ' Unicode nul terminated string
Const REG_DWORD = 4                      ' 32-bit number


Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"


Private Declare Function RegOpenKeyEx Lib "advapi32" Alias
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal
ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As
Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String,
ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String,
ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long)
As Long

Private Sub Form_Load()
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "."
& App.Revision
    lblTitle.Caption = App.Title
    lbLibraryVerison.Caption = "FM Control library version " +
fMainForm.Con.LibraryVersion     'Display Version of FM Control Library
End Sub

Private Sub cmdSysInfo_Click()
        Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
        Unload Me
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr


        Dim rc As Long
        Dim SysInfoPath As String
        ' Try To Get System Info Program Path\Name From Registry...
        If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From Registry...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
                ' Validate Existance Of Known 32 Bit File Version
                If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                        SysInfoPath = SysInfoPath & "\MSINFO32.EXE"


                ' Error - File Can Not Be Found...
                Else
                        GoTo SysInfoErr
                End If
        ' Error - Registry Entry Can Not Be Found...
        Else
                GoTo SysInfoErr
        End If


        Call Shell(SysInfoPath, vbNormalFocus)


        Exit Sub
SysInfoErr:
        MsgBox "System Information Is Unavailable At This Time",
vbOKOnly
End Sub


Public Function GetKeyValue(KeyRoot As Long, KeyName As String,
SubKeyRef As String, ByRef KeyVal As String) As Boolean
        Dim i As Long                                           ' Loop
Counter
        Dim rc As Long                                          ' Return
Code
        Dim hKey As Long                                        ' Handle
To An Open Registry Key
        Dim hDepth As Long                                      '
        Dim KeyValType As Long                                  ' Data
Type Of A Registry Key
        Dim tmpVal As String                                    '
Tempory Storage For A Registry Key Value
        Dim KeyValSize As Long                                  ' Size
Of Registry Key Variable
        '------------------------------------------------------------
        ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
        '------------------------------------------------------------
        rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) '
Open Registry Key


        If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError          ' Handle
Error...


        tmpVal = String$(1024, 0)                             ' Allocate
Variable Space
```

```
        KeyValSize = 1024                                           ' Mark
Variable Size


        '------------------------------------------------------------
        ' Retrieve Registry Key Value...
        '------------------------------------------------------------
        rc = RegQueryValueEx(hKey, SubKeyRef, 0, KeyValType, tmpVal,
KeyValSize)    ' Get/Create Key Value


        If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError             ' Handle
Errors


        tmpVal = VBA.Left(tmpVal, InStr(tmpVal, VBA.Chr(0)) - 1)
        '------------------------------------------------------------
        ' Determine Key Value Type For Conversion...
        '------------------------------------------------------------
        Select Case KeyValType                                      ' Search
Data Types...
        Case REG_SZ                                                 ' String
Registry Key Data Type
                KeyVal = tmpVal                                         '
Copy String Value
        Case REG_DWORD                                              ' Double
Word Registry Key Data Type
                For i = Len(tmpVal) To 1 Step -1                        '
Convert Each Bit
                        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
' Build Value Char. By Char.
                Next
                KeyVal = Format$("&h" + KeyVal)                         '
Convert Double Word To String
        End Select


        GetKeyValue = True                                          ' Return
Success
        rc = RegCloseKey(hKey)                                      ' Close
Registry Key
        Exit Function                                               ' Exit


GetKeyError:    ' Cleanup After An Error Has Occured...
        KeyVal = ""                                                 ' Set
Return Val To Empty String
        GetKeyValue = False                                         ' Return
Failure
        rc = RegCloseKey(hKey)                                      ' Close
Registry Key
End Function
```

```
VERSION 5.00
Object = "{BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.1#0"; "TABCTL32.OCX"
Begin VB.Form frmConfig
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Configuration"
   ClientHeight    =   4920
   ClientLeft      =   2565
   ClientTop       =   1500
   ClientWidth     =   5910
   Icon            =   "frmConfig.frx":0000
   KeyPreview      =   -1  'True
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   4920
   ScaleWidth      =   5910
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Begin TabDlg.SSTab tabConfig
      Height          =   4215
      Left            =   120
      TabIndex        =   13
      Top             =   120
      Width           =   5655
      _ExtentX        =   9975
      _ExtentY        =   7435
      _Version        =   393216
      Style           =   1
      Tabs            =   2
      TabsPerRow      =   2
      TabHeight       =   520
      TabCaption(0)   =   "Properties"
      TabPicture(0)   =   "frmConfig.frx":0442
      Tab(0).ControlEnabled=   -1  'True
      Tab(0).Control(0)=   "fraFrankingModes"
      Tab(0).Control(0).Enabled=   0   'False
      Tab(0).Control(1)=   "fraGeneral"
      Tab(0).Control(1).Enabled=   0   'False
      Tab(0).Control(2)=   "fraPrintImage"
      Tab(0).Control(2).Enabled=   0   'False
      Tab(0).Control(3)=   "fraBase"
      Tab(0).Control(3).Enabled=   0   'False
      Tab(0).Control(4)=   "fraFeatures"
      Tab(0).Control(4).Enabled=   0   'False
      Tab(0).ControlCount=   5
      TabCaption(1)   =   "Settings"
      TabPicture(1)   =   "frmConfig.frx":045E
      Tab(1).ControlEnabled=   0   'False
      Tab(1).Control(0)=   "fraLowCreditWarn"
      Tab(1).Control(1)=   "fraUserTimeout"
      Tab(1).Control(2)=   "fraKeyboard"
      Tab(1).Control(3)=   "fraHighValue"
      Tab(1).Control(4)=   "fraFrankMenu"
      Tab(1).Control(5)=   "fraDescReg"
      Tab(1).ControlCount=   6
      Begin VB.Frame fraLowCreditWarn
         Caption         =   "Low Credit warning"
         Height          =   1095
         Left            =   -72120
         TabIndex        =   53
         Top             =   480
   Width              =   2535
   Begin VB.CommandButton cmdLowCreditWarnDeactivate
      Caption         =   "Deactivate"
      Height          =   495
      Left            =   1320
      TabIndex        =   7
      Top             =   360
      Width           =   975
   End
   Begin VB.CommandButton cmdLowCreditWarnActivate
      Caption         =   "Activate"
      Height          =   495
      Left            =   240
      TabIndex        =   6
      Top             =   360
      Width           =   975
   End
End
Begin VB.Frame fraUserTimeout
   Caption            =   "User timeout"
   Height             =   1095
   Left               =   -72120
   TabIndex           =   52
   Top                =   2880
   Width              =   2535
   Begin VB.CommandButton cmdUserTimeOutDeactivate
      Caption         =   "Deactivate"
      Height          =   495
      Left            =   1320
      TabIndex        =   11
      Top             =   360
      Width           =   975
   End
   Begin VB.CommandButton cmdUserTimeOutActivate
      Caption         =   "Activate"
      Height          =   495
      Left            =   240
      TabIndex        =   10
      Top             =   360
      Width           =   975
   End
End
Begin VB.Frame fraKeyboard
   Caption            =   "FM keyboard"
   Height             =   1095
   Left               =   -74760
   TabIndex           =   51
   Top                =   2880
   Width              =   2055
   Begin VB.CommandButton cmdKeyboardUnlock
      Caption         =   "Unlock"
      Height          =   495
      Left            =   1080
      TabIndex        =   5
      Top             =   360
      Width           =   735
   End
   Begin VB.CommandButton cmdKeyboardLock
      Caption         =   "Lock"
      Height          =   495
      Left            =   240
      TabIndex        =   4
      Top             =   360
      Width           =   735
   End
End
Begin VB.Frame fraHighValue
   Caption         =   "High Value limit"
   Height          =   1095
   Left            =   -72120
   TabIndex        =   50
   Top             =   1680
   Width           =   2535
   Begin VB.CommandButton cmdHVDeactivate
      Caption         =   "Deactivate"
      Height          =   495
      Left            =   1320
      TabIndex        =   9
      Top             =   360
      Width           =   975
   End
   Begin VB.CommandButton cmdHVActivate
      Caption         =   "Activate"
      Height          =   495
      Left            =   240
      TabIndex        =   8
      Top             =   360
      Width           =   975
   End
End
Begin VB.Frame fraFeatures
   Caption         =   "Features"
   Height          =   1020
   Left            =   2880
   TabIndex        =   43
   Top             =   2160
   Width           =   2535
   Begin VB.Label lbNrMailClass
      Caption         =   "4"
      Height          =   255
      Left            =   1600
      TabIndex        =   49
      Top             =   720
      Width           =   900
   End
   Begin VB.Label lbMargin
      Caption         =   "Yes"
      Height          =   255
      Left            =   1600
      TabIndex        =   48
      Top             =   480
      Width           =   900
   End
   Begin VB.Label lbAutoDate
      Caption         =   "Yes"
      Height          =   255
      Left            =   1600
      TabIndex        =   47
      Top             =   240
      Width           =   900
   End
   Begin VB.Label lbNrMailClassLabel
```

```
         Caption         =    "Nr. of Mail Classes:"
         Height          =    255
         Left            =    120
         TabIndex        =    46
         Top             =    720
         Width           =    1400
      End
      Begin VB.Label lbMarginLable
         Caption         =    "Margin settable:"
         Height          =    255
         Left            =    120
         TabIndex        =    45
         Top             =    480
         Width           =    1400
      End
      Begin VB.Label lbAutoDateLabel
         Caption         =    "Automatic date:"
         Height          =    255
         Left            =    120
         TabIndex        =    44
         Top             =    240
         Width           =    1400
      End
   End
   Begin VB.Frame fraBase
      Caption         =    "Base"
      Height          =    1020
      Left            =    240
      TabIndex        =    38
      Top             =    2160
      Width           =    2535
      Begin VB.Label lbBaseSoftware
         Caption         =    "0000000B"
         Height          =    255
         Left            =    1000
         TabIndex        =    42
         Top             =    600
         Width           =    1500
      End
      Begin VB.Label lbBaseModel
         Caption         =    "B335P"
         Height          =    255
         Left            =    1000
         TabIndex        =    41
         Top             =    300
         Width           =    1500
      End
      Begin VB.Label lbSoftwareLabel
         Caption         =    "Software:"
         Height          =    255
         Left            =    120
         TabIndex        =    40
         Top             =    600
         Width           =    700
      End
      Begin VB.Label lbModelLabel
         Caption         =    "Model:"
         Height          =    255
         Left            =    120
         TabIndex        =    39
         Top             =    300
      Width           =   700
   End
End
Begin VB.Frame fraPrintImage
   Caption         =   "Print image"
   Height          =   1500
   Left            =   2880
   TabIndex        =   27
   Top             =   480
   Width           =   2535
   Begin VB.Label lbPrintImage
      Caption         =   "99999.00"
      Height          =   255
      Left            =   1500
      TabIndex        =   37
      Top             =   1200
      Width           =   1000
   End
   Begin VB.Label lbLastDecade
      Caption         =   "0..9"
      Height          =   255
      Left            =   1500
      TabIndex        =   36
      Top             =   960
      Width           =   1000
   End
   Begin VB.Label lbFixZeros
      Caption         =   "2"
      Height          =   255
      Left            =   1500
      TabIndex        =   35
      Top             =   720
      Width           =   1000
   End
   Begin VB.Label lbDecPos
      Caption         =   "2"
      Height          =   255
      Left            =   1500
      TabIndex        =   34
      Top             =   480
      Width           =   1000
   End
   Begin VB.Label lbNrDecades
      Caption         =   "5"
      Height          =   255
      Left            =   1500
      TabIndex        =   33
      Top             =   240
      Width           =   1000
   End
   Begin VB.Label lbPrintImageLabel
      Caption         =   "Print image:"
      Height          =   255
      Left            =   120
      TabIndex        =   32
      Top             =   1200
      Width           =   1300
   End
   Begin VB.Label lbLastDecadeLabel
      Caption         =   "Last decade type:"
      Height          =   255
      Left           =   120
      TabIndex       =   31
      Top            =   960
      Width          =   1300
   End
   Begin VB.Label lbFixZerosLabel
      Caption        =   "Nr. of fixed zeros:"
      Height         =   255
      Left           =   120
      TabIndex       =   30
      Top            =   720
      Width          =   1300
   End
   Begin VB.Label lbDecPosLabel
      Caption        =   "Decimal position:"
      Height         =   255
      Left           =   120
      TabIndex       =   29
      Top            =   480
      Width          =   1300
   End
   Begin VB.Label lbNrDecadesLabel
      Caption        =   "Nr. of decades:"
      Height         =   255
      Left           =   120
      TabIndex       =   28
      Top            =   240
      Width          =   1300
   End
End
Begin VB.Frame fraGeneral
   Caption        =   "General"
   Height         =   1500
   Left           =   240
   TabIndex       =   18
   Top            =   480
   Width          =   2535
   Begin VB.Label lbFMSoftware
      Caption        =   "JPAB005H"
      Height         =   255
      Left           =   1200
      TabIndex       =   26
      Top            =   720
      Width          =   1300
   End
   Begin VB.Label lbSerialNr
      Caption        =   "123456"
      Height         =   255
      Left           =   1200
      TabIndex       =   25
      Top            =   480
      Width          =   1300
   End
   Begin VB.Label lbFMType
      Caption        =   "F3XXPLUS"
      Height         =   255
      Left           =   1200
      TabIndex       =   24
      Top            =   240
      Width          =   1300
   End
```

```
      Begin VB.Label lbApplication
         Caption         =   "Auto Tax with Acquisition Tax"
         Height          =   495
         Left            =   1200
         TabIndex        =   23
         Top             =   960
         Width           =   1300
      End
      Begin VB.Label lbApplicationLabel
         Caption         =   "Application:"
         Height          =   255
         Left            =   120
         TabIndex        =   22
         Top             =   960
         Width           =   1000
      End
      Begin VB.Label lbFMSoftwareLabel
         Caption         =   "FM Software:"
         Height          =   255
         Left            =   120
         TabIndex        =   21
         Top             =   720
         Width           =   1000
      End
      Begin VB.Label lbSerialNrLabel
         Caption         =   "Serial Nr.:"
         Height          =   255
         Left            =   120
         TabIndex        =   20
         Top             =   480
         Width           =   1000
      End
      Begin VB.Label lbFMTypeLabel
         Caption         =   "FM Type:"
         Height          =   255
         Left            =   120
         TabIndex        =   19
         Top             =   240
         Width           =   1000
      End
   End
   Begin VB.Frame fraFrankingModes
      Caption         =   "Franking Modes"
      Height          =   600
      Left            =   2880
      TabIndex        =   16
      Top             =   3360
      Width           =   2535
      Begin VB.Label lbFrankModes
         Caption         =   "Normal / Tapes / Letters / Items"
         Height          =   255
         Left            =   120
         TabIndex        =   17
         Top             =   300
         Width           =   2380
      End
   End
   Begin VB.Frame fraFrankMenu
      Caption         =   "Franking menu"
      Height          =   1095
      Left            =   -74760
```

```
         TabIndex        =   15
         Top             =   1680
         Width           =   2055
         Begin VB.CommandButton cmdFrankMenuUnLock
            Caption         =   "Unlock"
            Height          =   495
            Left            =   1080
            TabIndex        =   3
            Top             =   360
            Width           =   735
         End
         Begin VB.CommandButton cmdFrankMenuLock
            Caption         =   "Lock"
            Height          =   495
            Left            =   240
            TabIndex        =   2
            Top             =   360
            Width           =   735
         End
      End
      Begin VB.Frame fraDescReg
         Caption         =   "Descending register"
         Height          =   1095
         Left            =   -74760
         TabIndex        =   14
         Top             =   480
         Width           =   2055
         Begin VB.CommandButton cmdDescRegHide
            Caption         =   "Hide"
            Height          =   495
            Left            =   1080
            TabIndex        =   1
            Top             =   360
            Width           =   735
         End
         Begin VB.CommandButton cmdDescRegShow
            Caption         =   "Show"
            Height          =   495
            Left            =   240
            TabIndex        =   0
            Top             =   360
            Width           =   735
         End
      End
   End
   Begin VB.CommandButton cmdClose
      Cancel          =   -1  'True
      Caption         =   "Close"
      Height          =   375
      Left            =   4680
      TabIndex        =   12
      Top             =   4455
      Width           =   1095
   End
End
Attribute VB_Name = "frmConfig"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
' frmConfig displays the properties of the connected
' Franking Machine and allows to change its settings.

Option Explicit

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdDescRegHide_Click()
    Dim Config As New FMCTRLLib.FMConfig        'Create an new FMConfig
object
On Error GoTo ErrorHandler:                     'Jump to ErrorHandler in
case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer
    Config.ActiveConnection = fMainForm.Con     'Define the connection
to be used by Config

    Config.DescRegHide                          'Hide Descending
register

    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    Set Config = Nothing                        'Disassociate Config
object
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    Set Config = Nothing                        'Disassociate Config
object
End Sub

Private Sub cmdDescRegShow_Click()              'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.DescRegShow                          'Show Descending
registier

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdFrankMenuLock_Click()            'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con
    Config.FrankMenuLock                        'Lock Franking menu

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdFrankMenuUnLock_Click()          'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.FrankMenuUnlock                      'Unlock Franking menu

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdHVActivate_Click()               'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.HVLimitActivate                      'Activate High Value
limit

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdHVDeactivate_Click()             'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.HVLimitDeactivate                    'Deactivate High Value
limit

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
```

```
ErrorHandler:
     Screen.MousePointer = vbDefault
     ErrorHandler
     Set Config = Nothing
End Sub

Private Sub cmdKeyboardLock_Click()               'See Sub
cmdDescRegHide_Click for commentary (same structure)
     Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
     Screen.MousePointer = vbHourglass
     Config.ActiveConnection = fMainForm.Con

     Config.KeyboardLock                          'Lock FM keyboard

     Screen.MousePointer = vbDefault
     Set Config = Nothing
     Exit Sub
ErrorHandler:
     Screen.MousePointer = vbDefault
     ErrorHandler
     Set Config = Nothing
End Sub

Private Sub cmdKeyboardUnlock_Click()             'See Sub
cmdDescRegHide_Click for commentary (same structure)
     Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
     Screen.MousePointer = vbHourglass
     Config.ActiveConnection = fMainForm.Con

     Config.KeyboardUnlock                        'Unlock FM keyboard

     Screen.MousePointer = vbDefault
     Set Config = Nothing
     Exit Sub
ErrorHandler:
     Screen.MousePointer = vbDefault
     ErrorHandler
     Set Config = Nothing
End Sub

Private Sub cmdLowCreditWarnActivate_Click()     'See Sub
cmdDescRegHide_Click for commentary (same structure)
     Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
     Screen.MousePointer = vbHourglass
     Config.ActiveConnection = fMainForm.Con

     Config.WarningLowCreditActivate              'Activate Low Credig
warning

     Screen.MousePointer = vbDefault
     Set Config = Nothing
     Exit Sub
ErrorHandler:
     Screen.MousePointer = vbDefault
     ErrorHandler
     Set Config = Nothing
End Sub
```

```
Private Sub cmdLowCreditWarnDeactivate_Click()  'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.WarningLowCreditDeactivate             'Deactivate Low Credit
warning

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdUserTimeOutActivate_Click()      'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.UserTimeoutActivate                    'Activate User Timeout

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub

Private Sub cmdUserTimeOutDeactivate_Click()    'See Sub
cmdDescRegHide_Click for commentary (same structure)
    Dim Config As New FMCTRLLib.FMConfig
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Config.ActiveConnection = fMainForm.Con

    Config.UserTimeoutDeactivate                  'Deactivate User Timeout

    Screen.MousePointer = vbDefault
    Set Config = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Config = Nothing
End Sub
```

```
VERSION 5.00
Begin VB.Form frmCounters
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Dialog Caption"
   ClientHeight    =   2520
   ClientLeft      =   2760
   ClientTop       =   3750
   ClientWidth     =   5775
   Icon            =   "frmCounters.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   2520
   ScaleWidth      =   5775
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Begin VB.Frame Frame2
      Caption         =   "Batch counter"
      Height          =   1695
      Left            =   2760
      TabIndex        =   5
      Top             =   120
      Width           =   2895
      Begin VB.CommandButton cmdBatchActivate
         Caption         =   "&Activate"
         Height          =   375
         Left            =   1920
         TabIndex        =   1
         Top             =   240
         Width           =   855
      End
      Begin VB.CommandButton cmdBatchClear
         Caption         =   "&Clear"
         Height          =   375
         Left            =   1920
         TabIndex        =   3
         Top             =   1200
         Width           =   855
      End
      Begin VB.CommandButton cmdBatchRead
         Caption         =   "&Read"
         Height          =   375
         Left            =   1920
         TabIndex        =   2
         Top             =   720
         Width           =   855
      End
      Begin VB.Label lbBatchItems
         Alignment       =   1  'Right Justify
         Caption         =   "4578"
         Height          =   255
         Left            =   840
         TabIndex        =   17
         Top             =   1200
         Width           =   855
      End
      Begin VB.Label lbBatchActiveLabel
         Caption         =   "Active:"
         Height          =   255
         Left            =   240
         TabIndex        =   16
```

```
          Top             =   480
          Width           =   495
       End
       Begin VB.Label lbBatchValue
          Alignment       =   1  'Right Justify
          Caption         =   "4567899"
          Height          =   255
          Left            =   840
          TabIndex        =   15
          Top             =   840
          Width           =   855
       End
       Begin VB.Label lbBatchActive
          Alignment       =   1  'Right Justify
          Caption         =   "Yes"
          Height          =   255
          Left            =   840
          TabIndex        =   14
          Top             =   480
          Width           =   855
       End
       Begin VB.Label lbBatchValueLabel
          Caption         =   "Value:"
          Height          =   255
          Left            =   240
          TabIndex        =   7
          Top             =   840
          Width           =   495
       End
       Begin VB.Label lbBatchItemsLabel
          Caption         =   "Items:"
          Height          =   255
          Left            =   240
          TabIndex        =   6
          Top             =   1200
          Width           =   495
       End
    End
    Begin VB.Frame Frame1
       Caption         =   "Postal counter"
       Height          =   1695
       Left            =   120
       TabIndex        =   4
       Top             =   120
       Width           =   2535
       Begin VB.Label lbItems
          Alignment       =   1  'Right Justify
          Caption         =   "9'999'999"
          Height          =   255
          Left            =   1200
          TabIndex        =   13
          Top             =   1200
          Width           =   1095
       End
       Begin VB.Label lbDescending
          Alignment       =   1  'Right Justify
          Caption         =   "9'999'999.00"
          Height          =   255
          Left            =   1200
          TabIndex        =   12
          Top             =   840
```

```
            Width           =   1095
        End
        Begin VB.Label lbAscending
            Alignment       =   1  'Right Justify
            Caption         =   "9'999'999.00"
            Height          =   255
            Left            =   1200
            TabIndex        =   11
            Top             =   480
            Width           =   1095
        End
        Begin VB.Label lbItemsLabel
            Caption         =   "Items:"
            Height          =   255
            Left            =   240
            TabIndex        =   10
            Top             =   1200
            Width           =   975
        End
        Begin VB.Label lbDescendingLabel
            Caption         =   "Descending:"
            Height          =   255
            Left            =   240
            TabIndex        =   9
            Top             =   840
            Width           =   975
        End
        Begin VB.Label lbAscendingLabel
            Caption         =   "Ascending:"
            Height          =   255
            Left            =   240
            TabIndex        =   8
            Top             =   480
            Width           =   975
        End
    End
    Begin VB.CommandButton cmdClose
        Cancel          =   -1  'True
        Caption         =   "Close"
        Height          =   375
        Left            =   4440
        TabIndex        =   0
        Top             =   2040
        Width           =   1215
    End
End
Attribute VB_Name = "frmCounters"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmCounters displays the current values of the counters
'and allows to control the Batch Counter of the FM.

Option Explicit

'Activates Batch Counter (switch it ON)

Private Sub cmdBatchActivate_Click()
    Dim Actions As New FMCTRLLib.FMActions      'Create an new FMActions
object
```

```
On Error GoTo ErrorHandler:                         'Jump to ErrorHandler in
case of an error
    Screen.MousePointer = vbHourglass               'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con        'Define the connection
to be used by Actions

    Actions.BatchCounterActivate                    'Activate the Batch
Counter in the FM

    Screen.MousePointer = vbDefault                 'Show default mouse
pointer
    Set Actions = Nothing                           'Disassociate Actions
object
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault                 'Show default mouse
pointer
    ErrorHandler                                    'Handle errors (show
appropriate message)
    Set Actions = Nothing                           'Disassociate Actions
object
End Sub

'Clears the Batch Counter and displays the counter values and state
(ON/OFF) before the clearing

Private Sub cmdBatchClear_Click()
    Dim Actions As New FMCTRLLib.FMActions          'Create an new FMActions
object
    Dim strFormatMask As String                     'Define needed variables
    Dim bOn As Boolean
    Dim nItems As Long
    Dim cValue As Currency
On Error GoTo ErrorHandler:                         'Jump to ErrorHandler in
case of an error
    Screen.MousePointer = vbHourglass               'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con        'Define the connection
to be used by Actions

    strFormatMask = CreateCurrencyFormatMask        'Create format mask for
displaying money values

    Actions.BatchCounterClear bOn, nItems, cValue   'Clear Batch Couter
and receive its values before the clearing

    lbBatchActive = BoolToYesNo(bOn)                'Display the last
counter values before clear
    lbBatchValue = Format(cValue, strFormatMask)
    lbBatchItems = Format(nItems, "##,##0")

    Screen.MousePointer = vbDefault                 'Show default mouse
pointer
    Set Actions = Nothing                           'Disassociate Actions
object
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault                 'Show default mouse
pointer
    ErrorHandler                                    'Handle errors (show
appropriate message)
    Set Actions = Nothing                          'Disassociate Actions
object
End Sub

'Reads the current counter values and state (ON/OFF)

Private Sub cmdBatchRead_Click()                   'See Sub
cmdBatchClear_Click for commentary (same structure)
    Dim Actions As New FMCTRLLib.FMActions
    Dim strFormatMask As String
    Dim bOn As Boolean
    Dim nItems As Long
    Dim cValue As Currency
On Error GoTo ErrorHandler:
    Screen.MousePointer = vbHourglass
    Actions.ActiveConnection = fMainForm.Con

    strFormatMask = CreateCurrencyFormatMask

    Actions.BatchCounterRead bOn, nItems, cValue    'Read Batch Counter
from FM

    lbBatchActive = BoolToYesNo(bOn)                'Display Batch
Couter values
    lbBatchValue = Format(cValue, strFormatMask)
    lbBatchItems = Format(nItems, "##,##0")

    Screen.MousePointer = vbDefault
    Set Actions = Nothing
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault
    ErrorHandler
    Set Actions = Nothing
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub
```

```
VERSION 5.00
Begin VB.Form frmDateTime
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Date & Time"
   ClientHeight    =   1680
   ClientLeft      =   2760
   ClientTop       =   3750
   ClientWidth     =   2535
   Icon            =   "frmDateTime.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   1680
   ScaleWidth      =   2535
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Begin VB.Frame fraDateTime
      Caption         =   "FM system date && time (local)"
      Height          =   975
      Left            =   120
      TabIndex        =   2
      Top             =   120
      Width           =   2295
      Begin VB.Label lbDateTime
         Alignment       =   2  'Center
         BackStyle       =   0  'Transparent
         Caption         =   "25.12.00  13:25:34"
         Height          =   255
         Left            =   120
         TabIndex        =   3
         Top             =   480
         Width           =   2055
      End
   End
   Begin VB.CommandButton cmdRefresh
      Caption         =   "&Refresh"
      Default         =   -1  'True
      Height          =   375
      Left            =   120
      TabIndex        =   0
      Top             =   1200
      Width           =   1095
   End
   Begin VB.CommandButton cmdClose
      Cancel          =   -1  'True
      Caption         =   "Close"
      Height          =   375
      Left            =   1320
      TabIndex        =   1
      Top             =   1200
      Width           =   1095
   End
End
Attribute VB_Name = "frmDateTime"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmDateTime displays the current local system time and date.
'The display can be updated by clicking a button.

Option Explicit

Private Sub cmdClose_Click()
    Unload Me
End Sub
```

```
'Read and display the current FM system time

Public Sub cmdRefresh_Click()
    Dim Actions As New FMCTRLLib.FMActions      'Create an new
FMActions object
On Error GoTo ErrorHandler:                     'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con    'Define the
connection to be used by Actions

    lbDateTime.Caption = Actions.GetTimeDate    'read Date and time
from FM

    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    Set Actions = Nothing                       'Disassociate Actions
object
    Exit Sub
ErrorHandler:
    Unload Me
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    Set Actions = Nothing                       'Disassociate Actions
object
End Sub
```

```
VERSION 5.00
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
"MSCOMCTL.OCX"
Begin VB.Form frmFranking
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Franking"
   ClientHeight    =   4440
   ClientLeft      =   2760
   ClientTop       =   3750
   ClientWidth     =   6975
   Icon            =   "frmFranking.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   4440
   ScaleWidth      =   6975
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Begin MSComctlLib.StatusBar sbStatusBar
      Align           =   2  'Align Bottom
      Height          =   255
      Left            =   0
      TabIndex        =   19
      Top             =   4185
      Width           =   6975
      _ExtentX        =   12303
      _ExtentY        =   450
      Style           =   1
      SimpleText      =   "Ready"
      _Version        =   393216
      BeginProperty Panels {8E3867A5-8586-11D1-B16A-00C0F0283628}
         NumPanels       =   1
         BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-00C0F0283628}
         EndProperty
      EndProperty
   End
   Begin VB.CommandButton cmdClearList
      Caption         =   "&Clear list"
      Height          =   375
      Left            =   5640
      TabIndex        =   9
      Top             =   2280
      Width           =   1215
   End
   Begin MSComctlLib.ListView lvFrankings
      Height          =   1815
      Left            =   120
      TabIndex        =   8
      TabStop         =   0   'False
      ToolTipText     =   "Frankings"
      Top             =   2280
      Width           =   5415
      _ExtentX        =   9551
      _ExtentY        =   3201
      View            =   3
      LabelWrap       =   0   'False
      HideSelection   =   -1  'True
      FullRowSelect   =   -1  'True
      GridLines       =   -1  'True
      _Version        =   393217
      ForeColor       =   -2147483640
      BackColor       =   -2147483643
      BorderStyle     =   1
      Appearance      =   1
      NumItems        =   5
      BeginProperty ColumnHeader(1) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
         Key             =    "Dummy"
         Text            =    "Dummy"
         Object.Width           =   0
      EndProperty
      BeginProperty ColumnHeader(2) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
         Alignment       =    1
         SubItemIndex    =    1
         Key             =    "Number"
         Text            =    "#"
         Object.Width           =   1058
      EndProperty
      BeginProperty ColumnHeader(3) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
         Alignment       =    1
         SubItemIndex    =    2
         Key             =    "Postage"
         Text            =    "Postage"
         Object.Width           =   1764
      EndProperty
      BeginProperty ColumnHeader(4) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
         Alignment       =    1
         SubItemIndex    =    3
         Key             =    "JobRest"
         Text            =    "Presel. rest"
         Object.Width           =   1764
      EndProperty
      BeginProperty ColumnHeader(5) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
         SubItemIndex    =    4
         Key             =    "Status"
         Text            =    "Status"
         Object.Width           =   3528
      EndProperty
   End
   Begin VB.CommandButton cmdGoQuiet
      Caption         =   "Go &Quiet"
      Height          =   375
      Left            =   5640
      TabIndex        =   7
      Top             =   720
      Width           =   1215
   End
   Begin VB.Frame fraSettings
      Caption         =   "Settings"
      Height          =   1815
      Left            =   120
      TabIndex        =   11
      Top             =   120
      Width           =   5415
      Begin VB.TextBox txtPreselectionDpt
         Alignment       =   1  'Right Justify
         Height          =   285
         Left            =   4200
         MaxLength       =   3
         TabIndex        =   5
         Text            =   "0"
         Top             =   1320
         Width           =   375
      End
      Begin VB.TextBox txtMargin
         Alignment       =   1  'Right Justify
         Height          =   285
         Left            =   4200
      MaxLength         =   3
      TabIndex          =   3
      Text              =   "20"
      Top               =   840
      Width             =   375
   End
   Begin VB.TextBox txtDptAcq
      Alignment         =   1  'Right Justify
      Height            =   285
      Left              =   4200
      TabIndex          =   1
      Text              =   "9'999'900"
      Top               =   360
      Width             =   975
   End
   Begin VB.TextBox txtPostage
      Alignment         =   1  'Right Justify
      Height            =   285
      Left              =   1560
      TabIndex          =   0
      Text              =   "9'999'900"
      Top               =   360
      Width             =   975
   End
   Begin VB.ComboBox cmbFrankMode
      Height            =   315
      ItemData          =   "frmFranking.frx":0442
      Left              =   1560
      List              =   "frmFranking.frx":0444
      Style             =   2  'Dropdown List
      TabIndex          =   4
      Top               =   1320
      Width             =   1215
   End
   Begin VB.ComboBox cmbMailClass
      Height            =   315
      Left              =   1560
      Style             =   2  'Dropdown List
      TabIndex          =   2
      Top               =   840
      Width             =   1215
   End
   Begin VB.Label lbPreselectionDpt
      Alignment         =   1  'Right Justify
      BackStyle         =   0  'Transparent
      Caption           =   "Preselection:"
      Height            =   255
      Left              =   2880
      TabIndex          =   17
      Top               =   1380
      Width             =   1215
   End
   Begin VB.Label lbMargin
      Alignment         =   1  'Right Justify
      BackStyle         =   0  'Transparent
      Caption           =   "Margin:"
      Height            =   255
      Left              =   2880
      TabIndex          =   16
      Top               =   900
      Width             =   1215
   End
   Begin VB.Label lbFrankMode
      Alignment         =   1  'Right Justify
      BackStyle         =   0  'Transparent
      Caption           =   "Franking Mode:"
```

```
         Height          =   255
         Left            =   240
         TabIndex        =   15
         Top             =   1380
         Width           =   1215
      End
      Begin VB.Label lbMailClass
         Alignment       =   1  'Right Justify
         BackStyle       =   0  'Transparent
         Caption         =   "Mail Class:"
         Height          =   255
         Left            =   240
         TabIndex        =   14
         Top             =   900
         Width           =   1215
      End
      Begin VB.Label lbDptAcq
         Alignment       =   1  'Right Justify
         BackStyle       =   0  'Transparent
         Caption         =   "Acquisition Tax:"
         Height          =   255
         Left            =   2880
         TabIndex        =   13
         Top             =   420
         Width           =   1215
      End
      Begin VB.Label lbPostage
         Alignment       =   1  'Right Justify
         BackStyle       =   0  'Transparent
         Caption         =   "Auto Tax:"
         Height          =   255
         Left            =   240
         TabIndex        =   12
         Top             =   420
         Width           =   1215
      End
   End
   Begin VB.CommandButton cmdClose
      Cancel          =   -1  'True
      Caption         =   "Close"
      Height          =   375
      Left            =   5640
      TabIndex        =   10
      Top             =   3720
      Width           =   1215
   End
   Begin VB.CommandButton cmdSetDecades
      Caption         =   "&Set Decades"
      Default         =   -1  'True
      Height          =   375
      Left            =   5640
      TabIndex        =   6
      Top             =   240
      Width           =   1215
   End
   Begin VB.Label lbFrankings
      BackStyle       =   0  'Transparent
      Caption         =   "Frankings:"
      Height          =   255
      Left            =   120
      TabIndex        =   18
      Top             =   2040
      Width           =   855
   End
End
Attribute VB_Name = "frmFranking"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmFranking provides the GUI for setting the FM ready for Franking.
'In addition it contains a list of the Frankings that have been
released.

Option Explicit

'Enable or disable the Preselection field depending on the chosen
'Franking Mode and set the field to a valid value.

Private Sub cmbFrankMode_Click()
    If cmbFrankMode.ItemData(cmbFrankMode.ListIndex) = 0 Then   'If
Franking Mode = Normal
        txtPreselectionDpt.Enabled = False
'Disable Preselection field
        txtPreselectionDpt.Text = 0
    Else
        txtPreselectionDpt.Enabled = True
        txtPreselectionDpt.Text = 3
    End If
End Sub

Private Sub cmdClearList_Click()
    lvFrankings.ListItems.Clear                  'Clear the Frankings
list
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

'Set FM to QUIET state (not ready for Franking)

Private Sub cmdGoQuiet_Click()
    Dim Actions As New FMCTRLLib.FMActions       'Create an new
FMActions object
On Error GoTo ErrorHandler:                      'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass            'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con     'Define the
connection to be used by Actions

    Actions.GoQuiet                              'Set FM to QUIET
state (not ready for Franking)
    frmFranking.sbStatusBar.SimpleText = "Quiet"    'Write to
StatusBar

    Screen.MousePointer = vbDefault              'Show default mouse
pointer
    Set Actions = Nothing                        'Disassociate Actions
object
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault              'Show default mouse
pointer
    ErrorHandler                                 'Handle errors (show
appropriate message)
    Set Actions = Nothing                        'Disassociate Actions
object
End Sub

'Set the FM ready for Franking
```

```
Private Sub cmdSetDecades_Click()
    Dim Actions As New FMCTRLLib.FMActions      'Create an new
FMActions object
    Dim Config As New FMCTRLLib.FMConfig        'Create an new
FMConfig object
    Dim Postage As Currency                     'Define needed
variables
    Dim AcquisitionTax As Currency
    Dim Department As Long
    Dim MailClass As Integer
    Dim FrankMode As FRANKMODES
    Dim Preselection As Integer
    Dim Margin As Integer
    Dim Ret As RETVALS
    Dim msg As String
On Error GoTo ErrorHandler:                     'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con    'Define the
connection to be used by Actions
    Config.ActiveConnection = fMainForm.Con     'Define the
connection to be used by Config

    If Config.AutoTaxFM Then                    'If FM is a special
Auto Tax application (Japan)
        Postage = CCur(txtPostage.Text)         'Convert the entered
values
        AcquisitionTax = CCur(txtDptAcq.Text)
        Department = CLng(txtPreselectionDpt.Text)
        Margin = CInt(txtMargin.Text)
        Ret = Actions.SetDecadesTax(Postage, AcquisitionTax,
Department, Margin)    'Set Decades for Auto Tax FM
    Else                                        'Standard application
        Postage = CCur(txtPostage.Text)         'Convert the entered
values
        Department = CLng(txtDptAcq.Text)
        MailClass = cmbMailClass.ListIndex + 1
        FrankMode = IndexToFrankMode(cmbFrankMode.ItemData
(cmbFrankMode.ListIndex)) 'Convert Franking Mode
        Preselection = CInt(txtPreselectionDpt.Text)
        Margin = CInt(txtMargin.Text)
        Ret = Actions.SetDecades(Postage, Department, MailClass,
FrankMode, Preselection, Margin)   'Set Decades for Standard FM
    End If

    ProcessRetVal Ret                           'Process the result
of the Set Decades command (show message if necessary)

    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    Set Actions = Nothing                       'Disassociate Actions
object
    Set Config = Nothing                        'Disassociate Config
object
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    Set Actions = Nothing                       'Disassociate Actions
object
    Set Config = Nothing                        'Disassociate Config
object
```

```
End Sub

'Prepare frmFranking for the application used by the connected FM.
'Some TextBoxes are used for different data entry depending on the
'application (Standard / Auto Tax)

Private Sub Form_Load()
    Dim Config As New FMCTRLLib.FMConfig        'Create an new
FMConfig object
    Dim MCTxts() As String                      'Define the needed
variables
    Dim i As Integer
On Error GoTo ErrorHandler                      'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer
    Config.ActiveConnection = fMainForm.Con     'Define the
connection to be used by Config

    txtPostage.Text = "0"                       'Initialize
Postage/AutoTax field
    txtDptAcq.Text = "0"                        'Initialize
Department/AcquisitionTax field
    If Config.AutoTaxFM = False Then            'If it is a Standard
application
        lbPostage.Caption = "Postage:"          'Label the fields
according to usage
        lbDptAcq.Caption = "Department:"
        lbPreselectionDpt.Caption = "Preselection:"

        If Config.SettableMailClasses > 0 Then          'If any Mail
Classes are available
            MCTxts = Config.MailClassTexts              'Read Mail
Class texts form FM
            cmbMailClass.Clear                          'Clear the
ComboBox
            For i = LBound(MCTxts) To UBound(MCTxts)    'Fill Mail
Class texts into the ComboBox
                cmbMailClass.AddItem MCTxts(i), i
            Next i
            cmbMailClass.ListIndex = 0                  'Initialize
Mail Class ComboBox (choose 1st entry)
        Else                                            'No
Mailclasses available
            cmbMailClass.Enabled = False                'Disable Mail
Class ComboBox
        End If

        cmbFrankMode.Clear
'Clear Mail Class ComboBox
        If Config.FrankModeAvailNorm Then                       'If
Franking Mode Normal is available
            cmbFrankMode.AddItem "Normal"
'Insert entry into ComboBox
            cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 0    'Add
item data to be able to identify the choosen entry
        End If
        If Config.FrankModeAvailTape Then                       'Same
as above for Franking Mode Tape
            cmbFrankMode.AddItem "Tapes"
            cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 1
        End If
        If Config.FrankModeAvailLetter Then                     'Same
as above for Franking Mode Letter
            cmbFrankMode.AddItem "Letters"
            cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 2
```

```
          End If
          If Config.FrankModeAvailItem Then                             'Same
 as above for Franking Mode Item
              cmbFrankMode.AddItem "Items"
              cmbFrankMode.ItemData(cmbFrankMode.NewIndex) = 3
          End If
          cmbFrankMode.ListIndex = 0
 'Initialize Mail Class ComboBox (choose 1st entry)
          cmbFrankMode.Enabled = cmbFrankMode.ListCount > 1
 'Enable Franking Mode ComboBox if there is more than one entry
          txtPreselectionDpt.MaxLength = 3
 'Limit the text length of the Preselection field to 3 (max = "999")
      Else     '(AutoTaxFM)                          'Auto Tax application
          lbPostage.Caption = "Auto Tax:"                'Label the fields
 according to usage
          lbDptAcq.Caption = "Acquisition Tax:"
          lbPreselectionDpt.Caption = "Department:"

          txtPreselectionDpt.Enabled = True                         'Enable
 the Department field
          txtPreselectionDpt.Width = 975                            'Set
 width of Department field (more space necessary than for
 Preselection)
          txtPreselectionDpt.MaxLength = 0                          'No text
 length limit
          txtDptAcq.Enabled = Config.AcquisitonTaxAvailable   'Enable
 Acquisition Tax field if application supports Acquisition Tax
          cmbMailClass.Enabled = False                              'Disable
 Mail Class field (Auto Tax does not support Mail Classes)
          cmbFrankMode.Enabled = False                              'Disable
 Franking Mode field (Auto Tax does not support Franking Modes)
      End If  '(AutoTaxFM)

      If Config.MarginAvailable Then                    'Is Margin settable
          txtMargin.Enabled = True                      'Enable TextBox
          txtMargin.Text = "20"                         'Set a valid value
      Else
          txtMargin.Enabled = False                     'Disable TextBox
          txtMargin.Text = "0"
      End If

      Screen.MousePointer = vbDefault                   'Show default mouse
 pointer
      Set Config = Nothing                              'Disassociate Config
 object
      Exit Sub
 ErrorHandler:
      Unload Me                                         'Close frmFranking
      Screen.MousePointer = vbDefault                   'Show default mouse
 pointer
      ErrorHandler                                      'Handle errors (show
 appropriate message)
      Set Config = Nothing                              'Disassociate Config
 object
 End Sub

 Private Sub Form_Unload(Cancel As Integer)
      If Len(fMainForm.Con.ConnectionString) <> 0 Then     'If
 connection exists
          cmdGoQuiet_Click                                 'Set FM to
 QUIET state (not ready for Franking)
      End If
 End Sub

 Private Sub txtDptAcq_GotFocus()
      SelectAll txtDptAcq                               'Select the whole
```

```
text in the TextBox
End Sub

Private Sub txtMargin_GotFocus()
    SelectAll txtMargin                          'Select the whole
text in the TextBox
End Sub

Private Sub txtPostage_GotFocus()
    SelectAll txtPostage                         'Select the whole
text in the TextBox
End Sub

Private Sub txtPreselectionDpt_GotFocus()
    SelectAll txtPreselectionDpt                 'Select the whole
text in the TextBox
End Sub
```

```
VERSION 5.00
Begin VB.Form frmInterface
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Interface"
   ClientHeight    =   1320
   ClientLeft      =   2760
   ClientTop       =   3750
   ClientWidth     =   2775
   Icon            =   "frmInterface.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   1320
   ScaleWidth      =   2775
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Begin VB.TextBox txtComPort
      Alignment       =   1  'Right Justify
      Height          =   285
      Left            =   1680
      MaxLength       =   2
      TabIndex        =   0
      Text            =   "1"
      Top             =   240
      Width           =   375
   End
   Begin VB.CommandButton cmdCancel
      Cancel          =   -1  'True
      Caption         =   "Cancel"
      Height          =   375
      Left            =   120
      TabIndex        =   2
      Top             =   840
      Width           =   1215
   End
   Begin VB.CommandButton cmdOK
      Caption         =   "OK"
      Default         =   -1  'True
      Height          =   375
      Left            =   1440
      TabIndex        =   1
      Top             =   840
      Width           =   1215
   End
   Begin VB.Label lbCOMPort
      Caption         =   "COM Port #:"
      Height          =   255
      Left            =   720
      TabIndex        =   3
      Top             =   300
      Width           =   975
   End
End
Attribute VB_Name = "frmInterface"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmInterface allows changing the COM Port that is used to
'connect to the FM.

Option Explicit

Private Sub cmdCancel_Click()
    Unload Me
End Sub
```

```
Private Sub cmdOK_Click()
On Error GoTo ErrorHandler:                             'Jump to ErrorHandler
in case of an error
    nComPort = CInt(txtComPort.Text)                    'Save COM Port # in
Global variable
    Unload Me                                           'Close the
frmInterface
    Exit Sub
ErrorHandler:
    ErrorHandler                                        'Handle errors (show
appropriate message - here usually conversion error)
    txtComPort.SetFocus
End Sub

Private Sub Form_Load()
    txtComPort.Text = CStr(nComPort)                    'Write current COM
Port # into TextBox
End Sub

Private Sub txtComPort_GotFocus()
    SelectAll txtComPort                                'Select the whole
text in the TextBox
End Sub
```

```
VERSION 5.00
Begin VB.Form frmMailClassAdjust
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Mail Class adjust"
   ClientHeight    =   1320
   ClientLeft      =   2760
   ClientTop       =   3750
   ClientWidth     =   3240
   Icon            =   "frmMailClassAdjust.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   1320
   ScaleWidth      =   3240
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Begin VB.ComboBox cmbMailClass
      Height          =   315
      ItemData        =   "frmMailClassAdjust.frx":0442
      Left            =   1680
      List            =   "frmMailClassAdjust.frx":0455
      Style           =   2  'Dropdown List
      TabIndex        =   0
      ToolTipText     =   "Printed Mail Class"
      Top             =   240
      Width           =   1455
   End
   Begin VB.CommandButton CancelButton
      Cancel          =   -1  'True
      Caption         =   "Close"
      Height          =   375
      Left            =   1920
      TabIndex        =   3
      Top             =   840
      Width           =   1215
   End
   Begin VB.CommandButton OKButton
      Caption         =   "&Adjust"
      Default         =   -1  'True
      Height          =   375
      Left            =   600
      TabIndex        =   1
      Top             =   840
      Width           =   1215
   End
   Begin VB.Label lbPrintedMC
      Alignment       =   1  'Right Justify
      BackStyle       =   0  'Transparent
      Caption         =   "Printed Mail Class:"
      Height          =   255
      Left            =   240
      TabIndex        =   2
      Top             =   300
      Width           =   1335
   End
End
Attribute VB_Name = "frmMailClassAdjust"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmMailClassAdjust allows correcting the Mail Class assignment to
'the Mail Class Cylinder position.
'In rare failure cases the assignment can get wrong. With this
'function the assignment can be adjusted.
```

```
Option Explicit

Private Sub CancelButton_Click()
    Unload Me
End Sub

'Executes Mail Class adjusting

Private Sub OKButton_Click()
    Dim Actions As New FMCTRLLib.FMActions      'Create an new
FMConfig object
    Dim MCTxts() As String                      'Define needed
variables
    Dim nMCCylPos As Integer
    Dim nEmptyPos As Integer
    Dim i As Integer
On Error GoTo ErrorHandler                      'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con    'Define the
connection to be used by Config

    If cmbMailClass.ListIndex < 0 Then          'If no Mail Class is
chosen
        MsgBox "Please choose a Mail Class.", vbExclamation     'Show
message
    Else
        Actions.MailClassAdjust cmbMailClass.ListIndex + 1
'Adjust Mail Class on FM
    End If

    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    Set Actions = Nothing                       'Disassociate Config
object
    Exit Sub
ErrorHandler:
    Unload Me
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    Set Actions = Nothing                       'Disassociate Config
object
End Sub
```

```
VERSION 5.00
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
"Mscomctl.ocx"
Begin VB.Form frmMain
   BackColor       =   &H80000005&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "FM Control User Application Demo"
   ClientHeight    =   4140
   ClientLeft      =   1410
   ClientTop       =   1410
   ClientWidth     =   7110
   Icon            =   "frmMain.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   ScaleHeight     =   4140
   ScaleWidth      =   7110
   StartUpPosition =   2  'CenterScreen
   Begin MSComctlLib.Toolbar tbToolBar
      Align           =   1  'Align Top
      Height          =   360
      Left            =   0
      TabIndex        =   1
      Top             =   0
      Width           =   7110
      _ExtentX        =   12541
      _ExtentY        =   635
      ButtonWidth     =   609
      ButtonHeight    =   582
      Appearance      =   1
      Style           =   1
      ImageList       =   "imlToolbarIcons"
      _Version        =   393216
      BeginProperty Buttons {66833FE8-8583-11D1-B16A-00C0F0283628}
         NumButtons      =   15
         BeginProperty Button1 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Key             =   "Interface"
            Object.ToolTipText     =   "Interface (Ctrl+I)"
            ImageKey        =   "Interface"
         EndProperty
         BeginProperty Button2 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Style           =   3
         EndProperty
         BeginProperty Button3 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Key             =   "Connect"
            Object.ToolTipText     =   "Connect (F3)"
            ImageKey        =   "Connect"
         EndProperty
         BeginProperty Button4 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Key             =   "Disconnect"
            Object.ToolTipText     =   "Disconnect (F4)"
            ImageKey        =   "Disconnect"
         EndProperty
         BeginProperty Button5 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Style           =   3
         EndProperty
         BeginProperty Button6 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Key             =   "Properties"
            Object.ToolTipText     =   "Properties (Ctrl+P)"
            ImageKey        =   "Properties"
         EndProperty
         BeginProperty Button7 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Key             =   "Settings"
            Object.ToolTipText     =   "Settings (Ctrl+S)"
            ImageKey        =   "Settings"
         EndProperty
         BeginProperty Button8 {66833FEA-8583-11D1-B16A-00C0F0283628}
```

```
            Style            =    3
         EndProperty
         BeginProperty Button9 {66833FEA-8583-11D1-B16A-00C0F0283628}
            Key              =    "Franking"
            Object.ToolTipText      =    "Franking (F5)"
            ImageKey         =    "Franking"
         EndProperty
         BeginProperty Button10 {66833FEA-8583-11D1-B16A-
00C0F0283628}
            Key              =    "Counters"
            Object.ToolTipText      =    "Counters (F6)"
            ImageKey         =    "Counters"
         EndProperty
         BeginProperty Button11 {66833FEA-8583-11D1-B16A-
00C0F0283628}
            Key              =    "DateTime"
            Object.ToolTipText      =    "Date && Time (F7)"
            ImageKey         =    "DateTime"
         EndProperty
         BeginProperty Button12 {66833FEA-8583-11D1-B16A-
00C0F0283628}
            Key              =    "MailClass"
            Object.ToolTipText      =    "Mail Class adjust (F8)"
            ImageKey         =    "MailClass"
         EndProperty
         BeginProperty Button13 {66833FEA-8583-11D1-B16A-
00C0F0283628}
            Key              =    "TextToDisplay"
            Object.ToolTipText      =    "Text to display (F9)"
            ImageKey         =    "TextToDisplay"
         EndProperty
         BeginProperty Button14 {66833FEA-8583-11D1-B16A-
00C0F0283628}
            Style            =    3
         EndProperty
         BeginProperty Button15 {66833FEA-8583-11D1-B16A-
00C0F0283628}
            Key              =    "About"
            Object.ToolTipText      =    "About"
            ImageKey         =    "Help"
         EndProperty
      EndProperty
      MouseIcon       =    "frmMain.frx":0442
   End
   Begin MSComctlLib.StatusBar sbStatusBar
      Align           =    2  'Align Bottom
      Height          =    255
      Left            =    0
      TabIndex        =    0
      Top             =    3885
      Width           =    7110
      _ExtentX        =    12541
      _ExtentY        =    450
      Style           =    1
      SimpleText      =    "Ready"
      _Version        =    393216
      BeginProperty Panels {8E3867A5-8586-11D1-B16A-00C0F0283628}
         NumPanels       =    3
         BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-00C0F0283628}
            AutoSize         =    1
            Object.Width           =    7355
            Text             =    "Status"
            TextSave         =    "Status"
         EndProperty
         BeginProperty Panel2 {8E3867AB-8586-11D1-B16A-00C0F0283628}
            Style            =    6
```

```
               AutoSize        =   2
               TextSave        =   "08.09.2000"
            EndProperty
            BeginProperty Panel3 {8E3867AB-8586-11D1-B16A-00C0F0283628}
               Style           =   5
               AutoSize        =   2
               TextSave        =   "16:29"
            EndProperty
         EndProperty
      End
      Begin MSComctlLib.ImageList imlToolbarIcons
         Left            =   6480
         Top             =   480
         _ExtentX        =   1005
         _ExtentY        =   1005
         BackColor       =   16777215
         ImageWidth      =   16
         ImageHeight     =   16
         MaskColor       =   12632256
         _Version        =   393216
         BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
            NumListImages   =   11
            BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":0B94
               Key             =   "Help"
            EndProperty
            BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":0CF2
               Key             =   "Counters"
            EndProperty
            BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":100C
               Key             =   "MailClass"
            EndProperty
            BeginProperty ListImage4 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":145E
               Key             =   "Franking"
            EndProperty
            BeginProperty ListImage5 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":18B0
               Key             =   "DateTime"
            EndProperty
            BeginProperty ListImage6 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":1D02
               Key             =   "Properties"
            EndProperty
            BeginProperty ListImage7 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":2154
               Key             =   "TextToDisplay"
            EndProperty
            BeginProperty ListImage8 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":25A6
               Key             =   "Connect"
            EndProperty
            BeginProperty ListImage9 {2C247F27-8591-11D1-B16A-
   00C0F0283628}
               Picture         =   "frmMain.frx":2704
               Key             =   "Disconnect"
         EndProperty
         BeginProperty ListImage10 {2C247F27-8591-11D1-B16A-
00C0F0283628}
            Picture         =   "frmMain.frx":2862
            Key             =   "Interface"
         EndProperty
         BeginProperty ListImage11 {2C247F27-8591-11D1-B16A-
00C0F0283628}
            Picture         =   "frmMain.frx":2974
            Key             =   "Settings"
         EndProperty
      EndProperty
   End
   Begin VB.PictureBox picAscomLogo
      Appearance      =   0  'Flat
      BackColor       =   &H80000005&
      BorderStyle     =   0  'None
      FillColor       =   &H00FFFFFF&
      ForeColor       =   &H80000005&
      Height          =   592
      Left            =   2520
      Picture         =   "frmMain.frx":2A86
      ScaleHeight     =   585
      ScaleWidth      =   2010
      TabIndex        =   2
      Top             =   3300
      Width           =   2010
   End
   Begin VB.Label lbSerialNr
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "654321"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   375
      Left            =   3480
      TabIndex        =   6
      Top             =   1680
      Width           =   1455
   End
   Begin VB.Label lbSerialNrLabel
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Serial Nr.:"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   375
      Left            =   2160
      TabIndex        =   5
      Top             =   1680
      Width           =   1215
   End
```

```
Begin VB.Label lbFMType
   BackColor       =   &H80000005&
   BackStyle       =   0  'Transparent
   Caption         =   "F3XXPLUS"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   375
   Left            =   3480
   TabIndex        =   4
   Top             =   1200
   Width           =   1455
End
Begin VB.Label lbFMTypeLabel
   BackColor       =   &H80000005&
   BackStyle       =   0  'Transparent
   Caption         =   "FM Type:"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   375
   Left            =   2160
   TabIndex        =   3
   Top             =   1200
   Width           =   1215
End
Begin VB.Menu mnuSystem
   Caption         =   "&System"
   Begin VB.Menu mnuSystemInterface
      Caption         =   "&Interface..."
      Shortcut        =   ^I
   End
   Begin VB.Menu mnuSystemBar0
      Caption         =   "-"
   End
   Begin VB.Menu mnuFileExit
      Caption         =   "E&xit"
   End
End
Begin VB.Menu mnuConnection
   Caption         =   "&Connection"
   Begin VB.Menu mnuConnectionConnect
      Caption         =   "&Connect"
      Shortcut        =   {F3}
   End
   Begin VB.Menu mnuConnectionDisconnect
      Caption         =   "&Disconnect"
      Shortcut        =   {F4}
   End
End
Begin VB.Menu mnuConfig
   Caption         =   "C&onfiguration"
   Begin VB.Menu mnuConfigProperties
      Caption         =   "&Properties..."
```

```
         Shortcut        =   ^P
      End
      Begin VB.Menu mnuConfigSettings
         Caption         =   "&Settings..."
         Shortcut        =   ^S
      End
   End
   Begin VB.Menu mnuActions
      Caption         =   "&Actions"
      Begin VB.Menu mnuActionsFranking
         Caption         =   "&Franking..."
         Shortcut        =   {F5}
      End
      Begin VB.Menu mnuActionsCounters
         Caption         =   "&Counters..."
         Shortcut        =   {F6}
      End
      Begin VB.Menu mnuActionsBar0
         Caption         =   "-"
      End
      Begin VB.Menu mnuActionsDateTime
         Caption         =   "&Date && Time..."
         Shortcut        =   {F7}
      End
      Begin VB.Menu mnuActionsMailClassAdjust
         Caption         =   "&Mail Class adjust..."
         Shortcut        =   {F8}
      End
      Begin VB.Menu mnuActionsTextToDisp
         Caption         =   "&Text to display..."
         Shortcut        =   {F9}
      End
   End
   Begin VB.Menu mnuView
      Caption         =   "&View"
      Begin VB.Menu mnuViewToolbar
         Caption         =   "&Toolbar"
         Checked         =   -1  'True
      End
      Begin VB.Menu mnuViewStatusBar
         Caption         =   "Status &Bar"
         Checked         =   -1  'True
      End
   End
   Begin VB.Menu mnuHelp
      Caption         =   "&Help"
      Begin VB.Menu mnuHelpAbout
         Caption         =   "&About..."
      End
   End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmMain is the main window of this application.
'It contains a menu and a toolbar that allow to establish connection,
'disconnect and open the sub forms.
'frmMain contains the Connection object that is used to comunicate
with the FM.
'Therefore also all events raised by the FM Control Library
(FMCTRLLib) are
'processed in this form.

Option Explicit
```

```
Public WithEvents Con As FMCTRLLib.Connection   'Define Connection
object variable
Attribute Con.VB_VarHelpID = -1

Public Enum FORMSTATES       'Enumerator for the different states of
the frmMain
    fsConnected
    fsDisconnected
End Enum

Dim nFrankingCounter As Long         'Counter to number the Frankings
Private bDieCoverOpen As Boolean     'Flag indicating whether the Die
Cover is open

'Event, raised whenever the Die Cover is closed

Private Sub Con_OnDieCoverClosed()
    bDieCoverOpen = False                               'Reset flag
    sbStatusBar.SimpleText = "Connected"                'Write to
statusbar of frmMain
    frmFranking.sbStatusBar.SimpleText = "Ready"        'Write to
statusbar of frmFranking (does not need to be open)
End Sub

'Event, raised whenever the Die Cover is closed

Private Sub Con_OnDieCoverOpen()
    bDieCoverOpen = True                                        'Set flag
    sbStatusBar.SimpleText = "Die Cover open"                   'Write to
statusbar of frmMain
    frmFranking.sbStatusBar.SimpleText = "Die Cover open"       'Write to
statusbar of frmFranking (does not need to be open)
End Sub

'Event, raised whenever an unexpected disconnection is detected

Private Sub Con_OnDisconnect()
    Dim i As Integer

    SetFormState (fsDisconnected)            'Set frmMain all controls
on frmMain for state Disconnected
    For i = Forms.Count - 1 To 1 Step -1     'Close all sub forms
        Unload Forms(i)
    Next
End Sub

'Event, raised whenever FM goes to QUIET state without having
received an explicit command to do so.
'This happens e.g. when the user timeout has run down.

Private Sub Con_OnQuiet(ByVal StatusCode As FMCTRLLib.RETVALS)
    If Not bDieCoverOpen Then                               'Do not
overwrite StatusBar if Die Cover is open
        frmFranking.sbStatusBar.SimpleText = "Quiet"        'Write to
statusbar of frmFranking (does not need to be open)
    End If
    If Not (StatusCode = R_OK) Then
        ProcessRetVal (StatusCode)                          'Display a
message for with the message from the high speed base
    End If
End Sub

'Event, raised whenever a Franking is released

Private Sub Con_OnFranking(ByVal Postage As Currency, ByVal JobRest
As Integer, ByVal StatusCode As FMCTRLLib.RETVALS)
    Dim liItem As ListItem                      'Define the needed
valiables
    Dim strJobRest As String

    nFrankingCounter = nFrankingCounter + 1
'Increment the Franking counter
    Set liItem = frmFranking.lvFrankings.ListItems.Add(1)
'Create a new list item on 1st position in the Frankings list on
frmFranking

'Column 0 is not used (1st column must be left alligned, but # should
be right

'alligned, so column 0 has width 0)
    liItem.SubItems(1) = CStr(nFrankingCounter)
'Write # to list item
    liItem.SubItems(2) = Format(Postage, CreateCurrencyFormatMask)
'Write formated Postage to list item
    If JobRest = -1 Then                        'If no Preselection
(-1)
        strJobRest = "(none)"
    Else
        strJobRest = CStr(JobRest)
    End If
    liItem.SubItems(3) = strJobRest                         'Write
remaining number of items to list item
    liItem.SubItems(4) = RetValToShortText(StatusCode)      'Write
short description of state to list item

    ProcessRetVal (StatusCode)                  'Process StatusCode
(show message if necessary)
End Sub

'Event, raised when no more Tapes are available in Franking Mode
FRK_TAPE

Private Sub Con_OnNoMoreTapes()
    Dim Actions As New FMCTRLLib.FMActions      'Create an new
FMActions object
    Dim msg As String                           'Define needed
variables
    Dim MsgBoxStyle As VbMsgBoxStyle
On Error GoTo ErrorHandler:                     'Jump to ErrorHandler
in case of an error
    Actions.ActiveConnection = fMainForm.Con    'Define the
connection to be used by Actions

    msg = "No more Tapes. Please refill."       'Prepare MsgBox text
    MsgBoxStyle = vbExclamation + vbRetryCancel 'MsgBox has Retry and
Cancel button

    If vbRetry = MsgBox(msg, MsgBoxStyle) Then  'Show message if
Retry button is pressed
        Screen.MousePointer = vbHourglass       'Show hourglass mouse
pointer
        Actions.TapesPresent                    'Tell FM that tapes
are ready again
    Else                                        'Cancel button
pressed
        Screen.MousePointer = vbHourglass       'Show hourglass mouse
pointer
        Actions.GoQuiet                         'Set FM to Quiet
state (not ready for Franking)
        frmFranking.sbStatusBar.SimpleText = "Quiet"    'Write to
statusbar of frmFranking (does not need to be open)
```

```
    End If

    Screen.MousePointer = vbDefault              'Show default mouse
pointer
    Set Actions = Nothing                        'Disassociate Actions
object
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault              'Show default mouse
pointer
    ErrorHandler                                 'Handle errors (show
appropriate message)
    Set Actions = Nothing                        'Disassociate Actions
object
End Sub

'Event, raised whenever a rotor error occurs

Private Sub Con_OnRotorError(ByVal StatusCode As FMCTRLLib.RETVALS)
    Dim msg As String

    sbStatusBar.SimpleText = "Connected"             'Write to
statusbar of frmMain
    frmFranking.sbStatusBar.SimpleText = "Quiet"     'Write to
statusbar of frmFranking (does not need to be open)

    msg = "Rotor error!" + Chr(13)               'Prepare message
    msg = msg + RetValToText(StatusCode)
    MsgBox msg, vbCritical                       'Display message
End Sub

Private Sub Form_Load()
    'frmMain is opende centred on the screen
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = 7200                              'Define width of
frmMain
    Me.Height = 4800                             'Define hight of
frmMain

    Set Con = New FMCTRLLib.Connection           'Create Connection
object
    SetFormState (fsDisconnected)                'Set frmMain all
controls on frmMain for state Disconnected
    sbStatusBar.SimpleText = "Ready"             'Write to statusbar
of frmMain
    nComPort = 1                                 'Initialize COM Port
number
End Sub

'Called before frmMain is closed

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As
Integer)
    If Con.ConnectionString <> "" Then           'If connection exists
        Cancel = True                            'Do not close frmMain
                                                 'Show message
        MsgBox "Can not exit while connection exists." + Chr(13) +
"Please disconnect.", vbCritical
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Dim i As Integer

    For i = Forms.Count - 1 To 1 Step -1         'Close all sub forms
```

```
        Unload Forms(i)
    Next
    If Me.WindowState <> vbMinimized Then      'Save window position
in registry
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
    End If
    Set Con = Nothing                          'Disassociate
Connection object
End Sub

'Information is filled into the fields and window is opended.

Private Sub mnuActionsCounters_Click()
    Dim Actions As New FMCTRLLib.FMActions     'Create an new
FMActions object
    Dim strFormatMask As String                'Define needed
variables
    Dim cAscending As Currency
    Dim cDescending As Currency
    Dim nItems As Long
    Dim bOn As Boolean
    Dim cValue As Currency
On Error GoTo ErrorHandler:                    'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass          'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con   'Define the
connection to be used by Actions

    strFormatMask = CreateCurrencyFormatMask   'Create format mask
for displaying money values

    Actions.GetCounterValues cAscending, cDescending, nItems    'Read
postal counters form FM

    With frmCounters
        .lbAscending.Caption = Format(cAscending, strFormatMask)
'Display postal counters
        .lbDescending.Caption = Format(cDescending, strFormatMask)
        .lbItems.Caption = Format(nItems, "##,##0")

        Actions.BatchCounterRead bOn, nItems, cValue    'Read Batch
Counter from FM

        .lbBatchActive = BoolToYesNo(bOn)               'Display
Batch Counter values
        .lbBatchValue = Format(cValue, strFormatMask)
        .lbBatchItems = Format(nItems, "##,##0")
    End With

    Screen.MousePointer = vbDefault            'Show default mouse
pointer
    Set Actions = Nothing                      'Disassociate Actions
object

    frmCounters.Show vbModal, Me

    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault            'Show default mouse
pointer
    ErrorHandler                               'Handle errors (show
appropriate message)
    Set Actions = Nothing                      'Disassociate Actions
object
```

```
End Sub

'Read and display the current FM system time and open window

Private Sub mnuActionsDateTime_Click()
    Dim Actions As New FMCTRLLib.FMActions      'Create an new
FMActions object
On Error GoTo ErrorHandler:                     'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer
    Actions.ActiveConnection = fMainForm.Con    'Define the
connection to be used by Actions

    frmDateTime.lbDateTime.Caption = Actions.GetTimeDate    'read
Date and time from FM

    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    Set Actions = Nothing                       'Disassociate Actions
object

    frmDateTime.Show vbModal, Me

    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    Set Actions = Nothing                       'Disassociate Actions
object
End Sub

'Prepare frmFranking for the application used by the connected FM.
'Some TextBoxes are used for different data entry depending on the
'application (Standard / Auto Tax)

Private Sub mnuActionsFranking_Click()
    Dim Config As New FMCTRLLib.FMConfig        'Create an new
FMConfig object
    Dim MCTxts() As String                      'Define the needed
variables
    Dim i As Integer
On Error GoTo ErrorHandler                      'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer
    Config.ActiveConnection = fMainForm.Con     'Define the
connection to be used by Config

    With frmFranking
        .txtPostage.Text = "0"                      'Initialize
Postage/AutoTax field
        .txtDptAcq.Text = "0"                       'Initialize
Department/AcquisitionTax field
        If Config.AutoTaxFM = False Then            'If it is a
Standard application
            .lbPostage.Caption = "Postage:"         'Label the fields
according to usage
            .lbDptAcq.Caption = "Department:"
            .lbPreselectionDpt.Caption = "Preselection:"

            If Config.SettableMailClasses > 0 Then          'If any
Mail Classes are available
                MCTxts = Config.MailClassTexts              'Read
```

```
 Mail Class texts form FM
                    .cmbMailClass.Clear                              'Clear
 the ComboBox
                    For i = LBound(MCTxts) To UBound(MCTxts)         'Fill
 Mail Class texts into the ComboBox
                        .cmbMailClass.AddItem MCTxts(i), i
                    Next i
                    .cmbMailClass.ListIndex = 0
 'Initialize Mail Class ComboBox (choose 1st entry)
               Else                                                  'No
 Mailclasses available
                    .cmbMailClass.Enabled = False                    'Disable
 Mail Class ComboBox
               End If

               .cmbFrankMode.Clear
 'Clear Mail Class ComboBox
               If Config.FrankModeAvailNorm Then
 'If Franking Mode Normal is available
                    .cmbFrankMode.AddItem "Normal"
 'Insert entry into ComboBox
                    .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 0
 'Add item data to be able to identify the choosen entry
               End If
               If Config.FrankModeAvailTape Then
 'Same as above for Franking Mode Tape
                    .cmbFrankMode.AddItem "Tapes"
                    .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 1
               End If
               If Config.FrankModeAvailLetter Then
 'Same as above for Franking Mode Letter
                    .cmbFrankMode.AddItem "Letters"
                    .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 2
               End If
               If Config.FrankModeAvailItem Then
 'Same as above for Franking Mode Item
                    .cmbFrankMode.AddItem "Items"
                    .cmbFrankMode.ItemData(.cmbFrankMode.NewIndex) = 3
               End If
               .cmbFrankMode.ListIndex = 0
 'Initialize Mail Class ComboBox (choose 1st entry)
               .cmbFrankMode.Enabled = .cmbFrankMode.ListCount > 1
 'Enable Franking Mode ComboBox if there is more than one entry
               .txtPreselectionDpt.MaxLength = 3
 'Limit the text length of the Preselection field to 3 (max = "999")
          Else      '(AutoTaxFM)                                 'Auto Tax
 application
               .lbPostage.Caption = "Auto Tax:"                      'Label the
 fields according to usage
               .lbDptAcq.Caption = "Acquisition Tax:"
               .lbPreselectionDpt.Caption = "Department:"

               .txtPreselectionDpt.Enabled = True
 'Enable the Department field
               .txtPreselectionDpt.Width = 975                           'Set
 width of Department field (more space necessary than for
 Preselection)
               .txtPreselectionDpt.MaxLength = 0                         'No
 text length limit
               .txtDptAcq.Enabled = Config.AcquisitonTaxAvailable
 'Enable Acquisition Tax field if application supports Acquisition Tax
               .cmbMailClass.Enabled = False
 'Disable Mail Class field (Auto Tax does not support Mail Classes)
               .cmbFrankMode.Enabled = False
 'Disable Franking Mode field (Auto Tax does not support Franking
 Modes)
```

```
          End If  '(AutoTaxFM)

          If Config.MarginAvailable Then                  'Is Margin
 settable
               .txtMargin.Enabled = True                  'Enable TextBox
               .txtMargin.Text = "20"                     'Set a valid
 value
          Else
               .txtMargin.Enabled = False                 'Disable TextBox
               .txtMargin.Text = "0"
          End If
     End With

     Screen.MousePointer = vbDefault                 'Show default mouse
 pointer
     Set Config = Nothing                            'Disassociate Config
 object

     frmFranking.Show vbModal, Me                    'Show window

     Exit Sub
 ErrorHandler:
     Screen.MousePointer = vbDefault                 'Show default mouse
 pointer
     ErrorHandler                                    'Handle errors (show
 appropriate message)
     Set Config = Nothing                            'Disassociate Config
 object
 End Sub

 'Prepare GUI for MailClassAdjust depending on the FM configuration
 and open window.

 Private Sub mnuActionsMailClassAdjust_Click()
     Dim Config As New FMCTRLLib.FMConfig            'Create an new
 FMConfig object
     Dim MCTxts() As String                          'Define needed
 variables
     Dim nMCCylPos As Integer
     Dim nEmptyPos As Integer
     Dim i As Integer
 On Error GoTo ErrorHandler                          'Jump to ErrorHandler
 in case of an error
     Screen.MousePointer = vbHourglass               'Show hourglass mouse
 pointer
     Config.ActiveConnection = fMainForm.Con         'Define the
 connection to be used by Config

     If Config.SettableMailClasses > 0 Then          'If Mail Classes are
 available
          nMCCylPos = Config.MailClassCylinderPositions   'Get number
 of positions on Mail Class cylinder from FM
          MCTxts = Config.MailClassTexts             'Get Mail Class texts
 from FM

          frmMailClassAdjust.cmbMailClass.Clear           'Clear ComboBox
          For i = LBound(MCTxts) To UBound(MCTxts)        'Fill Mail Class
 texts into ComboBox
               frmMailClassAdjust.cmbMailClass.AddItem MCTxts(i), i
          Next i

          nEmptyPos = nMCCylPos - (UBound(MCTxts) - LBound(MCTxts)) - 1
 'Calculate the number of positions on Mail

 'Class cylinder that are not assigned to a
```

```
'Mail Class text
        For i = 1 To nEmptyPos                                       'Fill
ComboBox with this number of "(no text)" entries
          frmMailClassAdjust.cmbMailClass.AddItem "(no text)"    'to
make all Mail Class cylinder positions available
        Next i

        Screen.MousePointer = vbDefault         'Show default mouse
pointer
        Set Config = Nothing                    'Disassociate Config
object

        frmMailClassAdjust.Show vbModal, Me     'Open the window
    Else                                        'No Mail Classes
available
        Screen.MousePointer = vbDefault         'Show default mouse
pointer
        Set Config = Nothing                    'Disassociate Config
object
        MsgBox "Mail Classes not available", vbCritical     'Show
message
    End If

    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    Set Config = Nothing                        'Disassociate Config
object
End Sub

Private Sub mnuActionsTextToDisp_Click()
    frmTextToDisplay.Show vbModal, Me
End Sub

Private Sub mnuConnectionConnect_Click()
    Dim strConnectionString As String
On Error GoTo ErrorHandler:                     'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass           'Show hourglass mouse
pointer

    strConnectionString = "COMPORT=" + CStr(nComPort) + ";
PROTOCOL=MLPV6"  'Compose ConnectionString

    Con.Connect (strConnectionString)           'Connect to FM

    nFrankingCounter = 0                        'Initialize Franking
Counter (used to number the Frankings)
    SetFormState (fsConnected)                  'Set frmMain all
controls on frmMain for state Connected
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    SetFormState (fsDisconnected)               'Set frmMain all
controls on frmMain for state Disconnected
End Sub

Private Sub mnuConnectionDisconnect_Click()
```

```
On Error GoTo ErrorHandler:                             'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass                   'Show hourglass mouse
pointer

    Con.Disconnect                                      'Disconnect from FM

    SetFormState (fsDisconnected)                       'Set frmMain all
controls on frmMain for state Disconnected
    Screen.MousePointer = vbDefault                     'Show default mouse
pointer
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault                     'Show default mouse
pointer
    ErrorHandler                                        'Handle errors (show
appropriate message) does usually not happen here
End Sub

'Executes appropriate mnuXxx_Click routine depending on the clicked
button

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComctlLib.Button)
On Error Resume Next
    Select Case Button.Key
    Case "Interface"
        mnuSystemInterface_Click
    Case "Connect"
        mnuConnectionConnect_Click
    Case "Disconnect"
        mnuConnectionDisconnect_Click
    Case "Properties"
        mnuConfigProperties_Click
    Case "Settings"
        mnuConfigSettings_Click
    Case "Franking"
        mnuActionsFranking_Click
    Case "Counters"
        mnuActionsCounters_Click
    Case "DateTime"
        mnuActionsDateTime_Click
    Case "MailClass"
        mnuActionsMailClassAdjust_Click
    Case "TextToDisplay"
        mnuActionsTextToDisp_Click
    Case "About"
        mnuHelpAbout_Click
    Case Else                                           'Unknown button
        MsgBox "Add ToolBar_ButtonClick code. " + Button.Key
    End Select
End Sub

Private Sub mnuConfigProperties_Click()
    LoadfrmConfig 0                                     'Display tab
"Properties"
End Sub

Private Sub mnuConfigSettings_Click()
    LoadfrmConfig 1                                     'Display tab
"Settings"
End Sub

'Information is filled into the fields and window is opended

Private Sub LoadfrmConfig(nTab As Integer)
    Dim Config As New FMCTRLLib.FMConfig                'Create an new
```

```
FMConfig object
On Error GoTo ErrorHandler:                              'Jump to ErrorHandler
in case of an error
    Screen.MousePointer = vbHourglass                    'Show hourglass mouse
pointer
    Config.ActiveConnection = fMainForm.Con              'Define the
connection to be used by Config

    With frmConfig
        'General information
        .lbFMType.Caption = FMTypeToString(Config.FMType)        'Read
and display FM Type
        .lbSerialNr.Caption = Config.SerialNr                    'Read
and display Serial Number
        .lbFMSoftware.Caption = Config.FMSWVer                   'Read
and display FM Software version
        If Config.AutoTaxFM Then                                 'If it is
a special Auto Tax app. (Japan)
            .lbApplication.Caption = "Auto Tax"                  'Compose
the description of the appliction
            If Config.AcquisitonTaxAvailable Then                'Does the
Auto Tax app. support Acquisition Tax?
                .lbApplication.Caption = .lbApplication.Caption + "
with Acquisition Tax"
            End If
        Else                                                     'Non Auto
Tax application
            .lbApplication.Caption = "Standard"
        End If

        'Print image
        .lbNrDecades.Caption = CStr(Config.DecadeNumber)         'Read
and display Number of Decade wheels
        .lbDecPos.Caption = CStr(Config.DecPointPosition)        'Read
and display position of the decimal point
        .lbFixZeros.Caption = CStr(Config.FixedZeros)            'Read
and display number of fixed zeros
        If Config.LastDecadeType = LD09 Then                     'Read and
display type of the last Decade wheel
            .lbLastDecade.Caption = "0..9"                       'Can be
all 0..9
        Else
            .lbLastDecade.Caption = "0/5"                        'Can
only be 0 or 5
        End If
        .lbPrintImage.Caption = CreatePrintImage                 'Compose
the print image (e.g. 99.99 or 9999900)

        'Information about the Base
        .lbBaseModel.Caption = BaseModelToString(Config.BaseModel)
'Read and display the Base model
        .lbBaseSoftware.Caption = Config.BaseSWVer
'Read and display Base software version

        'Features
        .lbAutoDate = BoolToYesNo(Config.AutoDate)
'Read and display whether FM has automatic date
        .lbMargin.Caption = BoolToYesNo(Config.MarginAvailable)
'Read and display whether margin is changable
        .lbNrMailClass.Caption = CStr(Config.SettableMailClasses)
'Read and display the number of available Mail Classes

        'Franking Modes
        .lbFrankModes = FrankModesToString                 'Compose a
String containing a list of the available Franking Modes
    End With
    Screen.MousePointer = vbDefault                'Show default mouse
pointer
    Set Config = Nothing                           'Disassociate Config
object

    frmConfig.tabConfig.Tab = nTab                 'Display the
appropriate tab
    frmConfig.Show vbModal, Me                     'Show the window

    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault                'Show default mouse
pointer
    ErrorHandler                                   'Handle errors (show
appropriate message)
    Set Config = Nothing                           'Disassociate Config
object
End Sub

Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub

Private Sub mnuSystemInterface_Click()
    frmInterface.Show vbModal, Me
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

Private Sub mnuViewToolbar_Click()
    mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
    tbToolBar.Visible = mnuViewToolbar.Checked
End Sub

Private Sub mnuFileExit_Click()
    Unload Me
End Sub

'Makes all the GUI elements appear in the appropriate way depending
on the FromState

Sub SetFormState(State As FORMSTATES)
    Dim Config As New FMCTRLLib.FMConfig          'Create an new
FMConfig object
On Error GoTo ErrorHandler:                       'Jump to ErrorHandler
in case of an error

    Select Case State
    Case fsConnected
        Config.ActiveConnection = Con             'Define the
connection to be used by Config
        lbFMType.Caption = FMTypeToString(Config.FMType)    'Read and
display FM Type
        lbSerialNr.Caption = Config.SerialNr      'Read and display
Serial number
        Set Config = Nothing                      'Disassociate Config
object

        sbStatusBar.SimpleText = "Connected"      'Write to status bar

        mnuSystemInterface.Enabled = False        'Enable or disable
the GUI elements
```

```
         mnuFileExit.Enabled = True
         mnuConnectionConnect.Enabled = False
         mnuConnectionDisconnect.Enabled = True
         mnuConfigProperties.Enabled = True
         mnuConfigSettings.Enabled = True
         mnuActionsFranking.Enabled = True
         mnuActionsCounters.Enabled = True
         mnuActionsDateTime.Enabled = True
         mnuActionsMailClassAdjust.Enabled = True
         mnuActionsTextToDisp.Enabled = True
         mnuViewToolbar.Enabled = True
         mnuViewStatusBar.Enabled = True
         mnuHelpAbout.Enabled = True
         With tbToolBar
             .Buttons("Interface").Enabled = False
             .Buttons("Connect").Enabled = False
             .Buttons("Disconnect").Enabled = True
             .Buttons("Properties").Enabled = True
             .Buttons("Settings").Enabled = True
             .Buttons("Franking").Enabled = True
             .Buttons("Counters").Enabled = True
             .Buttons("DateTime").Enabled = True
             .Buttons("MailClass").Enabled = True
             .Buttons("TextToDisplay").Enabled = True
             .Buttons("About").Enabled = True
         End With
     Case fsDisconnected
         lbFMType.Caption = "(none)"
         lbSerialNr.Caption = "(none)"
         sbStatusBar.SimpleText = "Disconnected" 'Write to status bar

         mnuSystemInterface.Enabled = True       'Enable or disable
 the GUI elements
         mnuFileExit.Enabled = True
         mnuConnectionConnect.Enabled = True
         mnuConnectionDisconnect.Enabled = False
         mnuConfigProperties.Enabled = False
         mnuConfigSettings.Enabled = False
         mnuActionsFranking.Enabled = False
         mnuActionsCounters.Enabled = False
         mnuActionsDateTime.Enabled = False
         mnuActionsMailClassAdjust.Enabled = False
         mnuActionsTextToDisp.Enabled = False
         mnuViewToolbar.Enabled = True
         mnuViewStatusBar.Enabled = True
         mnuHelpAbout.Enabled = True
         With tbToolBar
             .Buttons("Interface").Enabled = True
             .Buttons("Connect").Enabled = True
             .Buttons("Disconnect").Enabled = False
             .Buttons("Properties").Enabled = False
             .Buttons("Settings").Enabled = False
             .Buttons("Franking").Enabled = False
             .Buttons("Counters").Enabled = False
             .Buttons("DateTime").Enabled = False
             .Buttons("MailClass").Enabled = False
             .Buttons("TextToDisplay").Enabled = False
             .Buttons("About").Enabled = True
         End With
    End Select

    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault             'Show default mouse
pointer
    ErrorHandler                                'Handle errors (show
appropriate message)
    Set Config = Nothing                          'Disassociate Config
object
End Sub
```

```
VERSION 5.00
Begin VB.Form frmTextToDisplay
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Text to display"
   ClientHeight    =   1440
   ClientLeft      =   2760
   ClientTop       =   3750
   ClientWidth     =   5355
   Icon            =   "frmTextToDisplay.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   1440
   ScaleWidth      =   5355
   ShowInTaskbar   =   0   'False
   StartUpPosition =   1  'CenterOwner
   Begin VB.TextBox txtStartPos
      Alignment       =   1  'Right Justify
      Height          =   285
      Left            =   1200
      MaxLength       =   2
      TabIndex        =   1
      Text            =   "0"
      Top             =   840
      Width           =   375
   End
   Begin VB.TextBox txtText
      BeginProperty Font
         Name            =   "Fixedsys"
         Size            =   9
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   375
      Left            =   1200
      MaxLength       =   32
      TabIndex        =   0
      Text            =   " Ascom Mailsys     FM Control    "
      ToolTipText     =   "Text to display"
      Top             =   360
      Width           =   3975
   End
   Begin VB.CommandButton cmdClose
      Cancel          =   -1  'True
      Caption         =   "Close"
      Height          =   375
      Left            =   3960
      TabIndex        =   3
      Top             =   960
      Width           =   1215
   End
   Begin VB.CommandButton cmdSend
      Caption         =   "&Send"
      Default         =   -1  'True
      Height          =   375
      Left            =   2640
      TabIndex        =   2
      Top             =   960
      Width           =   1215
   End
   Begin VB.Label Label5
      Alignment       =   2  'Center
      BackStyle       =   0  'Transparent
```

```
         Caption         =   "16"
         Height          =   255
         Left            =   3000
         TabIndex        =   10
         Top             =   0
         Width           =   255
      End
      Begin VB.Line Line5
         X1              =   3120
         X2              =   3120
         Y1              =   240
         Y2              =   360
      End
      Begin VB.Label Label4
         Alignment       =   2  'Center
         BackStyle       =   0  'Transparent
         Caption         =   "30"
         Height          =   255
         Left            =   4680
         TabIndex        =   9
         Top             =   0
         Width           =   255
      End
      Begin VB.Label Label3
         Alignment       =   2  'Center
         BackStyle       =   0  'Transparent
         Caption         =   "20"
         Height          =   255
         Left            =   3480
         TabIndex        =   8
         Top             =   0
         Width           =   255
      End
      Begin VB.Label Label2
         Alignment       =   2  'Center
         BackStyle       =   0  'Transparent
         Caption         =   "10"
         Height          =   255
         Left            =   2280
         TabIndex        =   7
         Top             =   0
         Width           =   255
      End
      Begin VB.Label Label1
         Alignment       =   2  'Center
         BackStyle       =   0  'Transparent
         Caption         =   "1"
         Height          =   255
         Left            =   1200
         TabIndex        =   6
         Top             =   0
         Width           =   255
      End
      Begin VB.Line Line4
         X1              =   4780
         X2              =   4780
         Y1              =   240
         Y2              =   360
      End
      Begin VB.Line Line3
         X1              =   3600
         X2              =   3600
         Y1              =   240
         Y2              =   360
      End
      Begin VB.Line Line2
```

```
       X1                 =   2400
       X2                 =   2400
       Y1                 =   240
       Y2                 =   360
    End
    Begin VB.Line Line1
       X1                 =   1320
       X2                 =   1320
       Y1                 =   240
       Y2                 =   360
    End
    Begin VB.Label lbStartPos
       Alignment          =   1  'Right Justify
       Caption            =   "Start Position:"
       Height             =   255
       Left               =   120
       TabIndex           =   5
       Top                =   900
       Width              =   975
    End
    Begin VB.Label lbText
       Alignment          =   1  'Right Justify
       Caption            =   "Text:"
       Height             =   255
       Left               =   120
       TabIndex           =   4
       Top                =   420
       Width              =   975
    End
End
Attribute VB_Name = "frmTextToDisplay"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'frmTextToDisplay provides an interface to send a text to the
'FM that is written to its display

Option Explicit

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdSend_Click()
    Dim Actions As New FMCTRLLib.FMActions                 'Create an new
FMActions object
On Error GoTo ErrorHandler:                                'Jump to
ErrorHandler in case of an error
    Screen.MousePointer = vbHourglass                      'Show hourglass
mouse pointer
    Actions.ActiveConnection = fMainForm.Con               'Define the
connection to be used by Actions

    Actions.TextToDisplay txtText.Text, CInt(txtStartPos.Text)
'Display text on FM

    Screen.MousePointer = vbDefault                        'Show default
mouse pointer
    Set Actions = Nothing                                  'Disassociate
Actions object
    Exit Sub
ErrorHandler:
    Screen.MousePointer = vbDefault                        'Show default
mouse pointer
    ErrorHandler                                           'Handle errors
```

```
 (show appropriate message)
     Set Actions = Nothing                                  'Disassociate
Actions object
End Sub

Private Sub txtStartPos_GotFocus()
    SelectAll txtStartPos                                   'Select the whole
text in the TextBox
End Sub

Private Sub txtText_GotFocus()
    SelectAll txtText                                       'Select the whole
text in the TextBox
End Sub
```

```
Line 19: Class TabDlg.SSTab of control tabConfig was not a loaded
control class.
```

```
[SCC]
SCC=This is a source code control file
[FMCtrlDemo.vbp]
SCC_Project_Name=this project is not under source code control
SCC_Aux_Path=<This is an empty string for the mssccprj.scc file>
```

```
Type=Exe
Reference=*\G{00020430-0000-0000-C000-000000000046}#2.0#0#C:\WINNT
\System32\STDOLE2.TLB#OLE Automation
Reference=*\G{6B263850-900B-11D0-9484-00A0C91110ED}#1.0#0#C:\WINNT
\System32\MSSTDFMT.DLL#Microsoft Data Formatting Object Library
Reference=*\G{853A99F1-9752-11D3-819E-0050048C727A}#1.0#0#..
\FMCtrlLib\FMCtrl\Debug\FMCtrl_d.exe#FMCtrl 1.0 Type Library
Object={F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0; comdlg32.ocx
Object={831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0; Mscomctl.ocx
Object={BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.1#0; Tabctl32.ocx
Object={A8B3B723-0B5A-101B-B22E-00AA0037B2FC}#1.0#0; Grid32.OCX
Module=Start; Start.bas
Form=frmMain.frm
Form=frmAbout.frm
Form=frmConfig.frm
Form=frmInterface.frm
Form=frmCounters.frm
Form=frmDateTime.frm
Form=frmTextToDisplay.frm
Form=frmMailClassAdjust.frm
Form=frmFranking.frm
Module=ErrorHnd; ErrorHandler.bas
Module=Conversions; Conversions.bas
IconForm="frmMain"
Startup="Sub Main"
HelpFile=""
Title="FM Control User Application Demo"
ExeName32="FMCtrlDemo.exe"
Command32=""
Name="FMCtrlDemo"
HelpContextID="0"
Description="Demonstrates the usage of the FM Control library."
CompatibleMode="0"
MajorVer=1
MinorVer=2
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionComments="Demonstrates the usage of the FM Control library."
VersionCompanyName="Ascom Autelca AG Mailing Systems, Switzerland"
VersionFileDescription="Demonstrates the usage of the FM Control
library."
VersionProductName="FM Control User Application Demo"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FlPointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=1
DebugStartupOption=0
```

```
Start = 197, 107, 828, 629, C
frmMain = 33, 7, 903, 748, I, 267, 58, 792, 463, C
frmAbout = -2, 6, 789, 751, C, 205, 209, 662, 560, C
frmConfig = 191, 3, 927, 732, C, 266, 42, 723, 480, C
frmInterface = 49, 13, 848, 535, C, 15, 29, 246, 208, C
frmCounters = 258, 28, 846, 626, C, 379, 4, 813, 271, C
frmDateTime = 330, 97, 910, 671, C, 535, 21, 785, 240, C
frmTextToDisplay = 27, 17, 838, 693, C, 377, 7, 783, 203, C
frmMailClassAdjust = 84, 16, 915, 643, I, 470, 10, 781, 192, C
frmFranking = 12, 8, 799, 703, C, 180, 116, 691, 496, C
ErrorHnd = 9, 9, 897, 651, C
Conversions = 71, 2, 830, 735, C
```